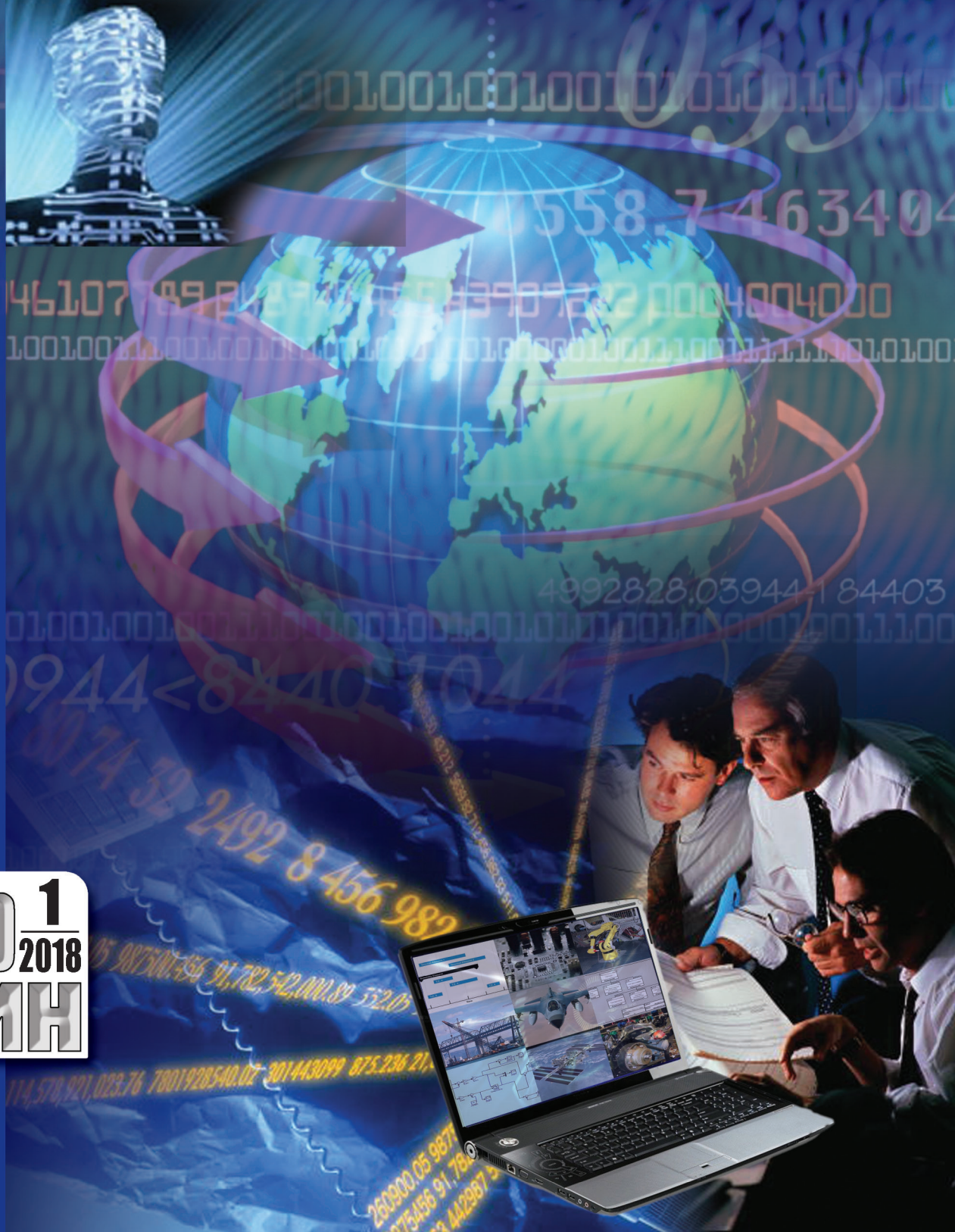


Программная инженерия



Пр^{**1**}
ИН₂₀₁₈
Том 9



Самарский национальный исследовательский университет имени академика С.П. Королева
Институт систем обработки изображений РАН – филиал федерального научно-исследовательского центра «Кристаллография и фотоника» Российской Академии Наук

ИНФОРМАЦИОННОЕ ПИСЬМО

О проведении IV Международной конференции и молодежной школы
«Информационные технологии и нанотехнологии» (ИТНТ - 2018)

ИТНТ-2018
itnt-conf.org

Цель Конференции:

предоставление ведущим ученым возможности научных дискуссий и обсуждения результатов фундаментальных и прикладных исследований в области информационных технологий и нанотехнологий; привлечение молодежи в сферу передовых научных исследований; обмен опытом научно-образовательной деятельности при подготовке ИТНТ-специалистов.

Тематика Конференции:

Секция 1 «Компьютерная оптика и нанофотоника»;

Секция 2 «Обработка изображений и дистанционное зондирование Земли»;

Секция 3 «Математическое моделирование физико-технических процессов и систем»;

Секция 4 «Науки о данных».

24 – 27
апреля
г. Самара,
Россия

Публикация работ:

Все представленные на Конференции работы будут опубликованы в сборнике трудов, индексируемом РИНЦ. По результатам Конференции отобранные материалы докладов будут рекомендованы к публикации на английском языке в изданиях, индексируемых Scopus и Web of Science.

Организационный взнос:

3000 руб.; студенты и аспиранты: 1000 руб.

Контакты:

Ответственный секретарь: secretary@itnt-conf.org

Руководство оргкомитета: steering@itnt-conf.org

Адрес: 443001, Россия, г. Самара, ул. Молодогвардейская 151

Программная инженерия

Том 9
№ 1
2018
Пр
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жиженко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

- Змеев О. А., Политов А. М., Цыганкова Я. М., Юровская А. С.**
Инструментальное средство управления вариантами использования разрабатываемого приложения 3
- Богоявленская О. Ю.** Распределенная многоагентная система мониторинга и прогнозирования производительности транспортного уровня сетей передачи данных 11
- Поршнев С. В., Пономарева О. А., Бородин А. М., Мирвода С. Г.**
Модификация функции penalty R-дерева над обобщенным деревом поиска индексов для повышения производительности модуля cube PostgreSQL 22
- Болотова С. Ю., Зонов А. В., Тютин А. П.** Навигация внутри помещений в мобильных приложениях 29
- Попович С. С.** Особенности автоматизации эксперимента и обработки результатов при исследовании теплообмена в сверхзвуковом потоке сжимаемого газа 35
- Итоги** тринадцатой конференции "Разработка ПО/CEE-SECR 2017" 46

Журнал зарегистрирован

в Федеральной службе

по надзору в сфере связи,

информационных технологий

и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индексы: по каталогу агентства "Роспечать" — 22765, по Объединенному каталогу "Пресса России" — 39795) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

Http://novtex.ru/prin/rus E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2018

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

Vol. 9

N 1

2018

Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.),
Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
ZHIZHCENKO A. B., Dr. Sci. (Phys.-Math.),
Acad. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad.
RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.),
Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.),
Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOV JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.),
Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R., Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBRJAKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A.V., CHUGUNOVA A.V.

CONTENTS

Zmeev O. A., Polotov A. M., Tsygankova Ya. M., Iurovskaya A. S.

A Tool for Managing the Use Cases of the Application Being
Developed 3

Bogoiavlenskaia O. V. The Distributed Multi-agent System
for Monitoring and Forecasting of the Networking Transport Layer
Performance 11

Porshnev S. V., Ponomareva O. A., Borodin A. M., Mirvoda S. G.

Improving Penalty Function of R-tree over Generalized Index Search
Tree as Possible Way to Advance Performance of PostgreSQL Cube
Extension 22

Bolotova S. Yu., Zonov A. V., Tutin A. P. Indoor Navigation
in Mobile Applications 29

Popovich S. S. Experimental Automation and Data Processing

Features for Supersonic Heat Transfer Research 35

CEE-SECR 2017 Conference Results 46

Information about the journal is available online at:
<http://novtex.ru/prin/eng> e-mail: prin@novtex.ru

О. А. Змеев, д-р физ.-мат. наук, профессор, e-mail: ozmeyev@gmail.com,
А. М. Политов, аспирант, e-mail: a.m.politov@gmail.com,
Я. М. Цыганкова, магистр, e-mail: chaykayana@gmail.com,
А. С. Юровская, аспирант, e-mail: yurovskaya.as@gmail.com,
Национальный исследовательский Томский государственный университет

Инструментальное средство управления вариантами использования разрабатываемого приложения

Объектом исследования, результаты которого представлены в статье, стали управление вариантами использования подлежащего разработке приложения и их преобразование в диаграммы анализа. Целью работы была разработка инструментария управления вариантами использования подлежащего разработке приложения, который бы поддерживал возможность работы с полной моделью вариантов использования, а также предоставлял функциональные возможности по преобразованию каждого сценария варианта использования в соответствующую модель анализа UML.

Ключевые слова: варианты использования, текстовые представления, графические представления, модель анализа, унифицированная модель, модули расширения

Введение

Одним из важнейших этапов практически любого процесса разработки программного обеспечения является рабочий процесс управления требованиями. Эта дисциплина обеспечивает соответствие функциональных возможностей системы ожиданиям заинтересованных сторон. В процессе управления требованиями к программной системе могут быть вовлечены различные участники — аналитики со стороны команды разработки, представители стороны заказчика, прочие заинтересованные в процессе разработки лица. Одним из самых распространенных способов представления требований в процессах разработки программного обеспечения, основанных на использовании объектно-ориентированного подхода, являются варианты использования подлежащего разработке приложения [1—3] (далее для краткости изложения — варианты использования).

Одним из преимуществ использования технологии представления требований, основанных на вариантах использования [4], является представление функциональных требований в виде текстового описания на естественном языке, для понимания которого не требуется знаний специализированных нотаций. Однако при этом нужно учитывать, что на основании вариантов использования происходят дальнейшие этапы разработки программного обеспечения (анализ, проектирование, тестирование). В таком случае для возможности автоматизированной обработки вариантов использования необходима некоторая формальная модель, которая, как правило,

отсутствует у текстовых представлений, но присутствует у графических.

На настоящее время существует ряд специализированных приложений, позволяющих моделировать функциональные требования к системам и специфицировать их в виде вариантов использования, а именно Microsoft Visio Studio, Visual Paradigm, Rational Rose Enterprise, aToucan [5—8]. Однако не все из перечисленных выше средств моделирования функциональных требований поддерживают механизм автоматического двустороннего преобразования между представлениями. Эти средства в целом предоставляют довольно ограниченный набор форматов специфицирования вариантов использования, который не во всех средствах можно расширить посредством добавления дополнительного модуля.

Целью работы, результаты которой представлены далее, являлась разработка инструментария управления вариантами использования, который бы поддерживал возможность работы как с полной моделью вариантов использования, так и с расширяемым набором представлений отдельного сценария варианта использования. Такой инструмент должен также предоставлять функциональные возможности для последующего преобразования сценария варианта использования, например в соответствующую модель анализа.

1. Используемые нотации

Данный раздел содержит информацию, необходимую для понимания сути работы.

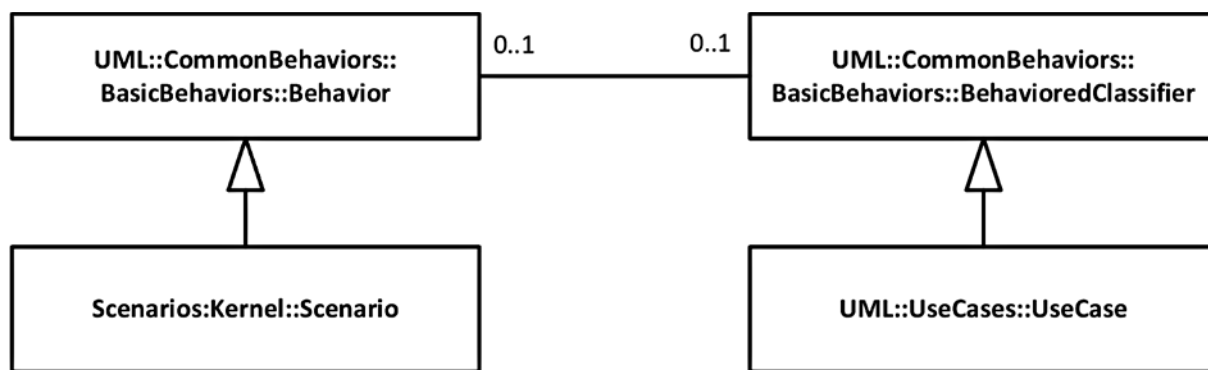


Рис. 1. Диаграмма классов, отображающая связь сценария варианта использования унифицированной модели с вариантом использования UML

1.1. Глоссарий

- *Вариант использования* (англ. *use case*) — описание последовательности действий, выполняемых системой и приносящих значимый результат конкретному действующему лицу (актеру). Вариант использования описывает, что делает система, но не указывает, каким образом она это делает [9].
- *Сценарий* — конкретная последовательность действий, иллюстрирующих поведение системы. Сценарий является экземпляром варианта использования.
- *Шаг сценария* — один из элементов последовательности действий, иллюстрирующих поведение системы.
- *Шаблон шага сценария* — структурный шаблон построения шага сценария, используемый для анализа и автоматического преобразования частей шага сценария в элементы диаграммы анализа.

1.2. Основа унифицированной модели вариантов использования

Существуют различные способы представления вариантов использования. Условно их можно разбить на следующие два вида:

- текстовое представление [10] — описывает последовательность действий на естественном языке с той или иной степенью формализма;
- графическое представление — описывает последовательность действий с помощью диаграмм, как правило, имеющих формальную модель.

Различные участники процесса разработки, задействованные в работе с вариантами использования, предпочитают использовать различные формы представления. Заказчики предпочитают использовать текстовые представления на естественном языке, поскольку для использования диаграмм необходимо знание модели и графической нотации [11].

Разработчики предпочитают использовать графические представления в виде диаграмм, поскольку такие представления имеют формальную модель,

которая может использоваться для дальнейших преобразований в другие необходимые артефакты.

Один из проблемных вопросов, который возникает в таком случае — отсутствие механизма автоматического преобразования между различными представлениями. В результате, в случаях, когда необходимы варианты использования в различных представлениях, нужно вручную создавать варианты использования в необходимых представлениях, и впоследствии выполнять синхронизацию изменений. Ответом на этот проблемный вопрос стало создание унифицированной модели вариантов использования, которая поддерживает существующие способы представления.

Унифицированная модель вариантов использования [12] представляет собой расширение к модели UML (версии 2.5), стандартизованной ISO [13, 14]. Основой для нее выступили пакеты UML::CommonBehaviors::BasicBehaviors::Behavior и UML::CommonBehaviors::BasicBehaviors. Для обеспечения совместимости с вариантами использования UML (UML::UseCases::UseCase) класс Scenario (Scenarios::Kernel), представляющий собой сценарий варианта использования в унифицированной модели, наследуется от класса Behavior (UML::CommonBehaviors::BasicBehaviors), с которым у вариантов использования есть ассоциация (рис. 1).

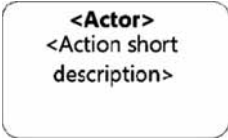
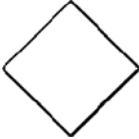
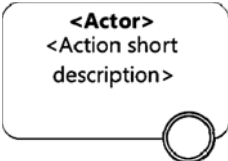
Таким образом, в дальнейшем описании будем использовать эту связь для работы с полной моделью вариантов использования.

1.3. Графическая нотация модели

Для визуального отображения предложенной модели в формальном графическом виде предложена графическая нотация, частично представленная в таблице. Данная нотация максимально приближена к элементам описания диаграмм состояния UML и знакома любому специалисту, применяющему этот язык на практике.

Пример сценария варианта использования в нотации унифицированной модели будет представлен ниже.

Графическая нотация унифицированной модели

Класс	Нотация	Описание нотации
ScenarioFlow		Представляется в виде стрелки, идущей из элемента source в элемент target. Если указано условие, то оно представляется в виде текста, расположенного рядом со стрелкой (ближе к ее началу)
Action		Представляется в виде прямоугольника со скругленными углами. Внутри прямоугольника на первой строке указывается актер, совершающий действие, он выделяется полужирным шрифтом. На последующих строках указывается краткое описание действия
Decision		Представляется в виде квадрата, повернутого на 45°
EndTrigger		Завершающий элемент представляется в виде окружности, внутри которой находится маркер, соответствующий типу события, или заполненная окружность, если событию не назначен никакой тип
StartHandler		Начало сценария представляется в виде заполненной окружности
BoundaryHandler		Представляется в виде двойной окружности, прикрепленной к границе действия, обработкой событий которого он занимается

2. Реализация трассировки в модель анализа

Изначально унифицированная модель вариантов использования [12] не поддерживала способа преобразования описанного варианта использования в соответствующую модель анализа. Для добавления этой возможности было предложено рассмотреть создание и применение шаблонов шага сценария варианта использования.

Если рассматривать шаг сценария варианта использования как объединение объектов предметной области, элементов пользовательского интерфейса предполагаемой системы и сопровождающего текста, то можно выделить довольно ограниченное число так называемых шаблонов шага сценария варианта использования. Примеры таких шаблонных шагов:

- актер кликает на {param1: некоторый элемент управления};
- система отображает уведомление {param1: некоторый текст};
- система делает активным {param1: некоторый элемент управления};
- актер вводит {param1: значение некоторого атрибута} в {param2: некоторый элемент управления};
- актер закрывает {param1: некоторое окно}.

На данных примерах видно, что в действиях задается сопровождающий текст "отображает уведомление", "делает активным", а также различные параметры в фигурных скобках, которые могут являться атрибутами объектов предметной области, элементами пользовательского интерфейса или фиксированным текстом.

Отобразим эту структуру на соответствующей диаграмме (рис. 2).

Класс StepTemplate необходим для указания шаблона шага, который может состоять из нескольких частей различного типа StepTemplatePart. Части шаблона шага задаются классами StepMetaAttributeParameter, StepMetaUIElementParameter, StepMetaConstTextParameter, каждый из которых определяет тип, объекты которого можно использовать при реализации шаблона. Также есть возможность задать сопровождающий текст, представленный классом StepAssociatedText.

Рассмотрим далее реализацию приведенных шаблонов шага уже с конкретными элементами пользовательского интерфейса, атрибутами объектов предметной области и фиксированным текстом:

- пользователь кликает на кнопку "Оформить заказ";
- система отображает уведомление "Платеж успешен!";

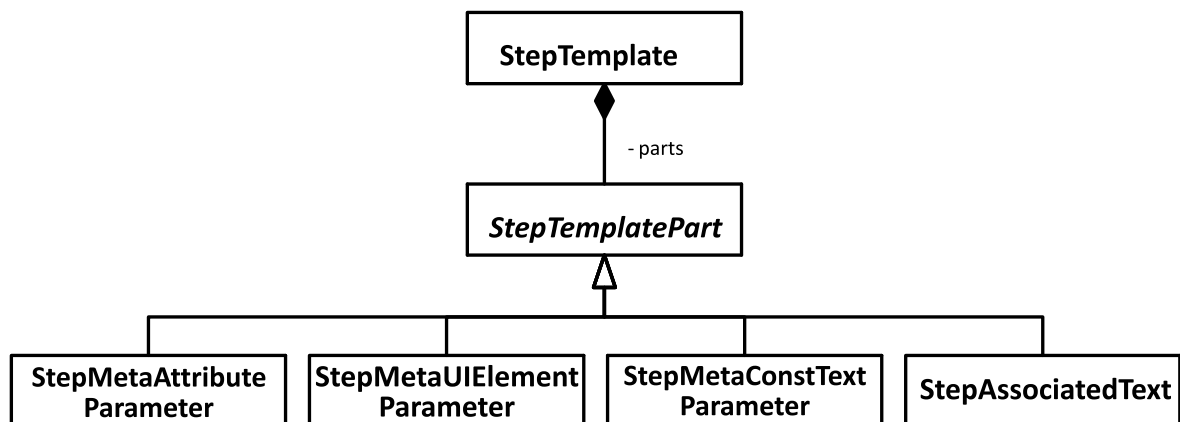


Рис. 2. Модель шаблона сценария варианта использования

- система делает активной кнопку "Оформить заказ";
- пользователь вводит адрес электронной почты в текстовое поле "email";
- пользователь закрывает окно "Справка".

В дальнейшем будем использовать вариант использования "Добавить в корзину", шаблоны шага для которого выглядят следующим образом:

- пользователь кликает на кнопку "Добавить в корзину";
- система проверяет число товаров в корзине;
- система открывает окно "Корзина";
- пользователь нажимает кнопку "Завершить покупку и перейти к оплате".

В перечисленных выше примерах есть действующее лицо — актер (пользователь, система), атрибуты объектов предметной области (тип оплаты, адрес электронной почты, товар), элементы пользовательского интерфейса (кнопка "Оформить заказ", кнопка "Завершить покупку и перейти к оплате", текстовое поле "email"), фиксированный текст ("Платеж успешен!") а также сопровождающий текст ("отображает уведомление", "делает активной", "проверяет количество", "открывает"). Причем некоторые шаблонные шаги можно ассоциировать не только с одним конкретным элементом пользовательского интерфейса, но и с другими.

Основываясь на перечисленных предположениях, можно спроектировать модель, представленную на рис. 3.

В данной модели классы с префиксом UCMN — классы, определенные в унифицированной модели и отвечающие за актера (UCMN.Actor), сценарий варианта использования в унифицированной модели (UCMN.Scenario) и шаг сценария (UCMN.ScenarioAction).

Класс StepTemplate нужен для указания шаблона шага, который

необходимо применить к этому действию, в то время как класс Step отображает уже конкретную реализацию шага сценария варианта использования с примененным шаблоном. Класс StepParameter показывает параметры какого типа и значения участвуют в этом шаге (рис. 4).

Класс StepAttributeParameter определяет значения части шаблона с типом StepMetaAttributeParameter, т. е. значение определенного атрибута из предметной области, StepUIElementParameter соотносит параметр шага типа StepMetaUIElementParameter с элементом пользовательского интерфейса, StepConstTextParameter задает конкретное значение для параметра с типом StepMetaConstTextParameter.

3. Архитектурное строение редактора

Для реализации инструментария управления вариантами использования была выбрана технология Windows Presentation Foundation (WPF) [13], которая используется для создания сложных пользователь-

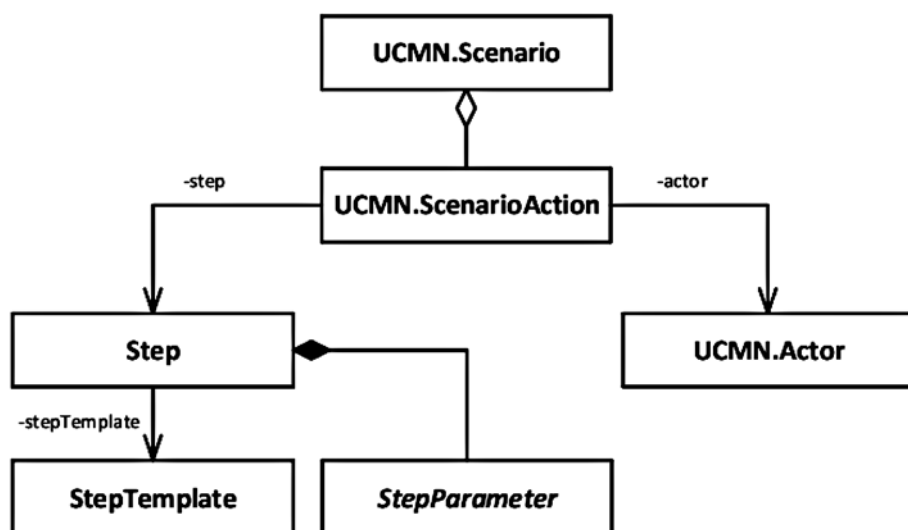


Рис. 3. Связь унифицированной модели и шаблона сценария варианта использования

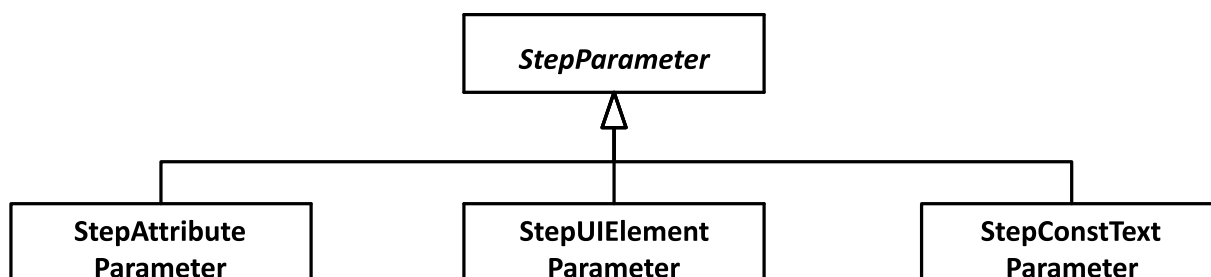


Рис. 4. Модель реализации шаблона шага сценария варианта использования

ских интерфейсов и высокопроизводительного отображения графических элементов.

Следуя рекомендации использования шаблона Модель—Представление—Модель представления (MVVM) [16] при разработке WPF-приложения, была выработана архитектурная модель приложения, представленная на рис. 5.

К основным пакетам редактора относятся перечисленные далее.

- Пакет UCGram.Application отвечает за отображение всего редактора в целом. Является частью, реализующей представление и модель представления в шаблоне MVVM.

- Пакет UCGram.Model содержит базовый класс для предоставления интерфейса уведомления и уникальной идентификации объектов внутри приложения.

- Пакет UCGram.Model.UseCases отражает предметную область, связанную с элементами диаграммы вариантов использования.

- Пакет UCGram.Model.UCMN отражает предметную область относительно графической нотации унифицированной модели вариантов использования.

- Пакет UCGram.Plugins содержит базовые классы для разработки плагинов, содержащих дополнительные способы представления спецификации сценария варианта использования.

- Пакет UCGram.PluginsManager содержит классы, отвечающие за получение информации о доступных плагинах и подключение этих плагинов в систему.

- Пакет UCGram.Synchronization содержит базовые классы для синхронизации представлений сценариев вариантов использования. Должен использоваться разработчиком для реализации механизма синхронизации разрабатываемых им представлений и представлений в нотации унифицированной модели.

- Пакет UCGram.Model.ScenarioStep содержит классы, моделирующие шаблоны шага сценария

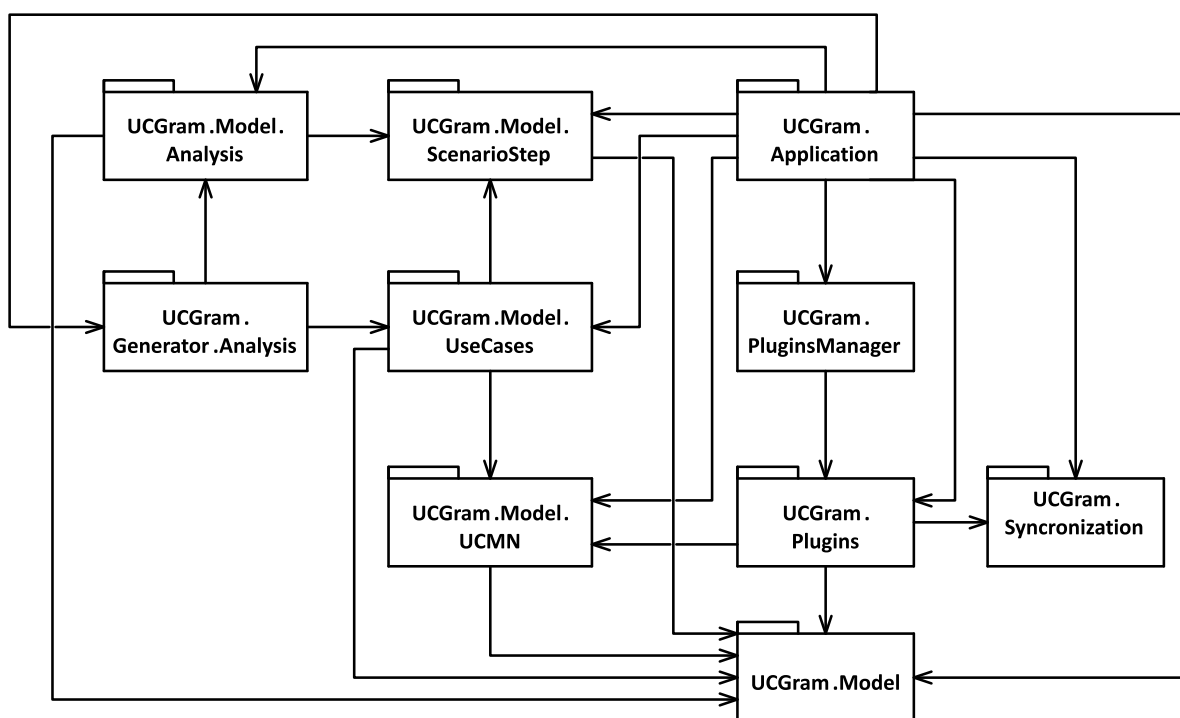


Рис. 5. Диаграмма пакетов ядра инструмента управления вариантами использования

варианта использования и их реализации. В его составе также содержатся модели объектов предметной области и элементов пользовательского интерфейса.

- Пакет UCGram.Model.Analysis содержит базовые классы для отображения и генерации диаграмм анализа.
- Пакет UCGram.Generator.Analysis содержит класс, необходимый для генерации диаграммы анализа на основании сценария варианта использования.

4. Пример работы реализованного инструмента

Следуя представленному подходу было реализовано и протестировано инструментальное средство управления вариантами использования, удовлетворяющее перечисленным выше функциональным требованиям. Рассмотрим подробнее механизм его работы с вариантами использования. Проект состоит из диаграмм вариантов использования, сценариев

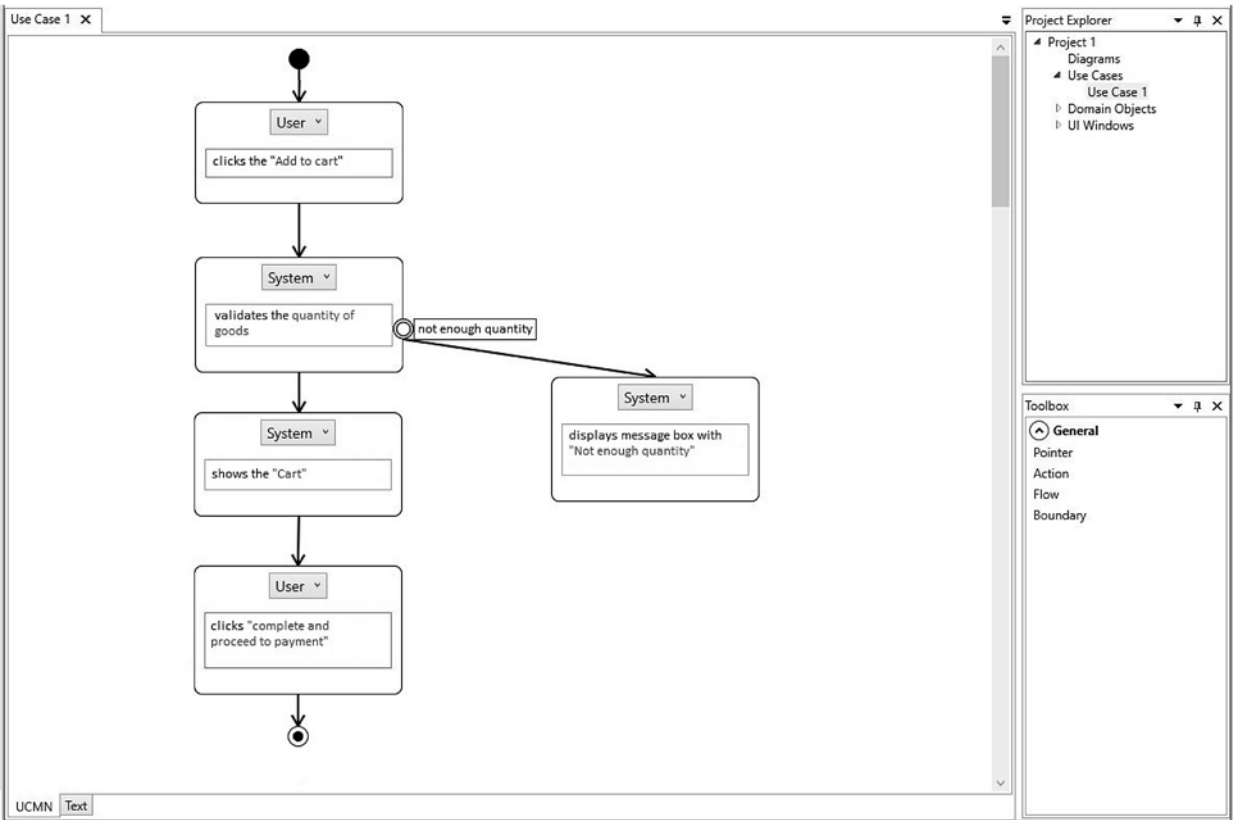


Рис. 6. Окно редактирования сценария варианта использования "Add to Cart" (Добавить в корзину) в унифицированной модели

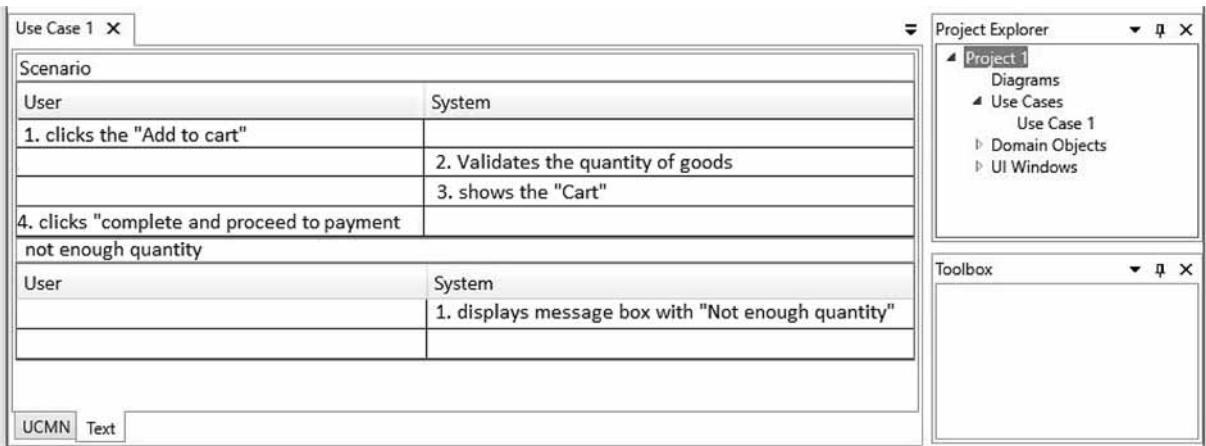


Рис. 7. Окно редактирования сценария варианта использования "Add to Cart" (Добавить в корзину) в текстовой модели

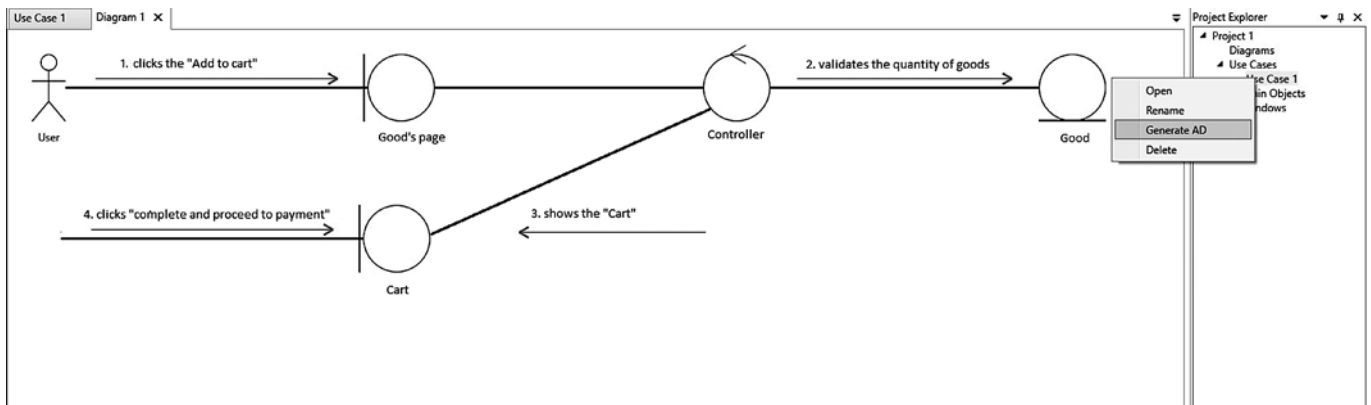


Рис. 8. Пример сгенерированной диаграммы анализа "Add to Cart" (Добавить в корзину)

вариантов использования, объектов предметной области и элементов пользовательского интерфейса, которые можно добавить к проекту.

Новый вариант использования создается посредством добавления его к проекту или при вызове спецификации варианта использования на диаграмме. В появившемся окне документа можно увидеть представление сценария варианта использования в нотации унифицированной модели. На рис. 6 представлен заданный сценарий варианта использования.

В текстовом варианте данный сценарий проиллюстрирован на рис. 7.

Следует учесть, что для возможности корректного построения диаграммы анализа (рис. 8) при редактировании шага сценария варианта использования требуется выбирать шаблон, который подходит для описания шага.

Заключение

В ходе работы были рассмотрены текущие средства, позволяющие в том или ином виде специфицировать варианты использования. Проанализированы их достоинства и недостатки, которые были учтены при реализации инструментального средства управления вариантами использования.

На основе выявленных функциональных требований было принято решение об использовании плагинов для расширения способов представления, позволяющих сторонним разработчикам для собственных нужд или по заданию своих организаций добавлять новый формат спецификаций и иметь возможность синхронизации его с существующими в системе. Также реализован плагин, предоставляющий реализацию двухколоночной нотации Р. Вирфс-Брок, относящийся к классу текстовых представлений. Таким образом была проверена работоспособность системы расширения редактора.

Для генерации модели анализа спроектировано и реализовано соответствующее расширение унифицированной модели, включающее в себя управление объектами предметной области и элементами поль-

зовательского интерфейса, а также правила генерации элементов модели анализа в виде элементов диаграммы UML и связей между ними.

Предложенное расширение унифицированной модели можно использовать для преобразования вариантов использования в артефакт последующих этапов разработки программного обеспечения, такой как модель тестирования.

Использованный в статье подход не позволяет решить все задачи, возникающие в дисциплине управления требованиями. Однако он позволяет поддерживать целостность формирования вариантов использования при работе с различными заинтересованными сторонами, задействованными при разработке программного обеспечения.

Список литературы

1. Jacobson I., Chirsterson M., Jonsson P., Overgaard G. Object Oriented Software Engineering: A Use Case Driven Approach: 1st Edition. ACM Press Staff. 1993. 552 p.
2. Jacobson I. Formalizing use-case modeling // JOOP. 1995. Vol. 8, No. 3. P. 10–14.
3. UML Use Case Diagrams: Guidelines. URL: <http://msdn.microsoft.com/en-us/library/dd409432.aspx>
4. Якобсон А., Буч Г., Рамбо Д. Унифицированный процесс разработки программного обеспечения. СПб: Питер, 2002. 496 с.
5. Software Design Tools for Agile Teams, with UML, BPMN and More. URL: <http://www.visual-paradigm.com/>
6. IBM — Rational Rose Enterprise. URL: <http://www-03.ibm.com/software/products/ru/enterprise>
7. Yue T., Briand L. C., Labiche Y. AToucan: an Automated Framework two Derive UML Analysis Models From Use Case Models//In ACM Transactions on Software Engineering and Methodology (TOSEM). 2015. Vol. 24, No. 3. P. 1–52.
8. Сунетов Т. Р., Шестаков Д. А., Свистунов И. В. Сравнение современных CASE-средств // Интерактивная наука. 2017. № 1. С. 142–144.
9. Cockburn A. Writing Effective Use Cases. 1st ed. Addison-Wesley Professional, 2000. 304 p.
10. Wirfs-Brock R. Designing Scenarios: Making the Case for a Use Case Framework // The Smalltalk Report. November-December 1993. Vol. 3, No. 3. URL: <http://www.wirfs-brock.com/PDFs/Designing%20Scenarios.pdf>
11. Jacobson I., Spence I., Bittner K. Use-Case 2.0 The Guide to Succeeding with Use Cases. URL: https://www.ivarjacobson.com/sites/default/files/field_iji_file/article/use-case_2_0_jan11.pdf

12. Змеев О. А., Политов А. М., Чайка Я. М. Концепция унифицированной модели сценария вариантов использования для фиксации функциональных требований к программному продукту // Вестник Том. Гос. ун-та. Управление, вычислительная техника и информатика. 2015. № 3 (32). С. 91–98.

13. ISO Object Management Group Unified Modeling Language (OMG UML). Part 1: Infrastructure. URL: <https://www.iso.org/standard/32624.html>

14. ISO Object Management Group Unified Modeling Language (OMG UML). Part 2: Superstructure. URL: <https://www.iso.org/standard/52854.html>

15. MSDN Magazine. URL: <http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>

16. Фаулер М. Архитектура корпоративных систем. М.: Вильямс, 2006. 404 с.

A Tool for Managing the Use Cases of the Application Being Developed

O. A. Zmeev, ozmeyer@gmail.com, A. M. Politov, a.m.politov@gmail.com,
Ya. M. Tsygankova, chaykayana@gmail.com, A. S. Iurovskaya, yurovskaya.as@gmail.com,
National Research Tomsk State University, Tomsk, 634050, Russian Federation

Corresponding author:

Zmeev Oleg A., Professor, National Research Tomsk State University, Tomsk, 634050, Russian Federation
E-mail: ozmeyer@gmail.com

*Received on October 16, 2017
Accepted on November 09, 2017*

The object of the study was the management of use cases and their transformation into analysis charts. The aim of the work was the development of a management tool for use cases that would support the ability to work with a full model of use cases and provide a functional transformation of the scenario of the use case in the corresponding UML analysis model.

Keywords: use cases, textual representations, graphical representations, analysis model, unified model, extension modules

For citation:

Zmeev O. A., Politov A. M., Tsygankova Ya. M., Iurovskaya A. S. A Tool for Managing the Use Cases of the Application Being Developed, *Programmnaya Ingeneria*, 2018, vol. 9, no. 1, pp. 3–10.

DOI: 10.17587/prin.9.3-10

References

1. Jacobson I., Chirsterson M., Jonsson P., Overgaard G. *Object Oriented Software Engineering: A Use Case Driven Approach*. 1st Edition, ACM Press Staff, 1993, 552 p.

2. Jacobson I. Modeling with Use Cases: Formalizing use-case modeling, *JOOP*, 1995, vol. 8, no. 3, pp. 10–14.

3. UML Use Case Diagrams: Guidelines, available at: <http://msdn.microsoft.com/en-us/library/dd409432.aspx>

4. Jacobson I., Booch G., Rumbaugh J. *Unified software development process*, Addison-Wesley Professional, 1999, 496 p.

5. Software Design Tools for Agile Teams, with UML, BPMN and More, available at: <http://www.visual-paradigm.com/>

6. IBM — Rational Rose Enterprise, available at: <http://www-03.ibm.com/software/products/ru/enterprise>

7. Yue T., Briand L. C., Labiche Y. AToucan: an Automated Framework two Derive UML Analysis Models From Use Case Models, *In ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2015, vol. 24, no. 3, pp. 1–52.

8. Sunetov T. R., Shestakov D. A., Svistunov I. V. Svrnenie sovremennyh CASE-sredstv (Comparison of modern CASE-tools), *Interaktivnaya nauka*, 2017, no 1, pp. 142–144 (in Russian).

9. Cockburn A. *Writing Effective Use Cases*: 1st ed. Addison-Wesley Professional, 2000, 304 p.

10. Wirfs-Brock R. Designing Scenarios: Making the Case for a Use Case Framework, *The Smalltalk Report*, November-December 1993, vol. 3, no. 3, available at: <http://www.wirfs-brock.com/PDFs/Designing%20Scenarios.pdf>

11. Jacobson I., Spence I., Bittner K. Use-Case 2.0 The Guide to Succeeding with Use Cases, available at: https://www.ivarjacobson.com/sites/default/files/field_iji_file/article/use-case_2_0_jan11.pdf

12. Zmeev O. A., Politov A. M., Chayka Ya. M. Konceptiya unificirovannoy modeli scenariya variantov ispol'zovaniya dlja fiksacii funkcional'nyh trebovanij k programnomu produktu (The concept of a unified scenario model for use cases for fixation functional requirements to a software product), *Vestn. Tom. Gos. un-ta. Upravlenie, vychislitel'naja tehnika i informatika*, 2015, no. 3 (32), pp. 91–98 (in Russian).

13. ISO Object Management Group Unified Modeling Language (OMG UML). Part 1: Infrastructure, available at: <https://www.iso.org/standard/32624.html>

14. ISO Object Management Group Unified Modeling Language (OMG UML). Part 2: Superstructure, available at: <https://www.iso.org/standard/52854.html>

15. MSDN Magazine, available at: <http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>

16. Fowler M. *Patterns of enterprise application architecture*, Addison-Wesley Professional, 2003, 533 p.

О. Ю. Богоявленская, канд. техн. наук, доц., e-mail: olbgvl@cs.karelia.ru,
ФГБОУ ВПО "Петрозаводский государственный университет"

Распределенная многоагентная система мониторинга и прогнозирования производительности транспортного уровня сетей передачи данных

Описана архитектура распределенной системы мониторинга и прогнозирования производительности сетей передачи данных. Система представляет собой совокупность агентов, осуществляющих мониторинг, анализ данных и прогнозирование производительности наиболее загруженных сетевых маршрутов, позволяющую пользователям и сетевым приложениям выбирать наиболее эффективные стратегии использования сетевых ресурсов. Приведены примерные сценарии использования, а также экспериментальный пример.

Ключевые слова: мониторинг, ядро ОС Linux, производительность сетей передачи данных, соединения транспортного уровня

Введение

Проблема управления производительностью сетей передачи данных приобрела особую актуальность уже на ранних стадиях становления метасети Интернет. По мере увеличения масштабов и с ростом диверсификации элементов сетевой инфраструктуры усилия по управлению и администрированию стали необходимым условием эффективного функционирования системы. В настоящее время эта проблема усугубляется рядом факторов, обусловленных технологическими решениями, принятыми на ранних этапах развития метасети. К их числу относятся, например, недостаточность сведений о загруженности сетевых маршрутов, которыми располагают источники данных, отсутствие у источников и получателей данных возможности выбора сетевого маршрута, недостаточная эффективность средств балансирования нагрузки, отсутствие функциональной иерархии систем управления. Как следствие, наличие крупных централизованных ресурсов (центры обработки данных, YouTube), а также потенциал перспективных парадигм сетевого управления, например, Information Centric Networks (ICN), Publish/Subscribe Internet (PSI) [1] в условиях отсутствия средств эффективного контроля и предотвращения перегрузок создают фундаментальные ограничения для дальнейшего развития сетей передачи данных [2].

В подходах к решению этой проблемы можно выделить два основных направления. Первое предполагает разработку и внедрение алгоритмов и протоколов передачи данных, а также новых методов организации сетей. Сюда можно отнести многочисленные алгоритмы улучшения производительности физического и канального уровней сети, алгоритмы

контроля перегрузки транспортного уровня. Некоторые из этих технологий широко известны. Это, например, методы обнаружения коллизий Ethernet, протокол четырехкратного рукопожатия и метод предотвращения коллизий протоколов IEEE 802.11x, алгоритмы предотвращения перегрузок протокола TCP и многие другие. Следует также упомянуть разнообразные технологии решения проблемы "последней мили", т. е. обеспечения соединения между индивидуальным пользователем и поставщиком услуг первого уровня, протоколы обеспечения качества услуг SLA, MPLS и др.

Второе направление подразумевает разработку и внедрение систем мониторинга и последующего анализа производительности сетей и их фрагментов, основанного на математических моделях и статистических методах. Здесь можно выделить системы пассивного и активного мониторинга. Такие системы перехватывают данные о трафике фрагмента сети, а также осуществляют первичный анализ и представление данных либо инициативно устанавливая тестовые соединения в целях проверки свойств сетевого окружения.

В настоящее время можно наблюдать тенденцию, в рамках которой появляются технологии, объединяющие оба подхода и предполагающие разработку средств управления базовыми элементами сети. Это, например, так называемые промежуточные устройства (*middleboxes*), которые изменяют свойства потоков данных и содержание управляющих структур пакетов в целях повышения безопасности соединений и улучшения ее производительности, например PEP [3], а также модели OpenFlow [4].

Достоинства и недостатки обоих подходов взаимосвязаны. Так, внедрение новых протоколов и алго-

ритмов часто улучшает производительность системы в целом, однако предполагает существенные усилия по их разработке, калибровке и последующему внедрению и не гарантирует обеспечения требований производительности отдельных сеансов и приложений. Влияние протоколов, предоставляющих гарантии качества отдельным соединениям, таким образом, ограничивается отдельным фрагментом сети, в котором реализованы соответствующие технологии. В свою очередь, системы мониторинга проводят анализ и предоставляют данные, которые позволяют администраторам улучшить работу и поддерживать уровень производительности корпоративных и/или локальных сетей. Однако администраторы таких сетей не могут влиять на элементы инфраструктуры, находящиеся вне зоны их ответственности. Поэтому даже при условии эффективной работы отдельного фрагмента сети, его пользователи могут быть разочарованы качеством предоставляемых услуг.

В контексте публикации предлагается адаптивный подход, который позволит снизить зависимость производительности процесса передачи данных от внешних факторов, а на основе результатов мониторинга вырабатывать обоснованные стратегии использования региональной и глобальной сетевой инфраструктуры. С этой целью разработана открытая архитектура распределенной системы, которая проводит мониторинг потоков, данных на транспортном уровне (уровень точка—точка) в ограниченном множестве источников и получателей данных; осуществляет анализ первичных данных; формирует прогноз производительности процесса передачи данных для отдельных источников и получателей данных. Транспортный уровень архитектуры OSI (Open System Interconnection) в рамках такого подхода выбран как основной объект анализа, так как он предоставляет интегральную характеристику производительности сетевого маршрута, которая отражает влияние многочисленных факторов, реализованных на нижележащих уровнях архитектуры. Заметим также, что именно производительность на транспортном уровне непосредственно вносит существенный вклад в оценку пользователем качества сетевых услуг.

Представляется, что такой подход к проблеме мониторинга позволит сочетать преимущества двух основных направлений подходов к решению проблем мониторинга и управления производительностью, описанных выше. Анализ наиболее активных сетевых маршрутов, основной сегмент которых находится за пределами локальной или корпоративной сети, позволяет косвенно контролировать состояние элементов сетевой инфраструктуры за пределами полномочий системных администраторов. В то же время открытая архитектура, предполагающая распределенное взаимодействие совокупности агентов, позволяет использовать сочетание различных систем и технологий. Она не требует структурных изменений больших фрагментов сети и допускает персонализированное использование. Прогнозы производительности, предоставляемые такой системой,

могут использоваться как отдельными пользователями в личных целях, в том числе и на мобильных устройствах, так и сетевыми приложениями, операторами связи, поставщиками контента, а также в контентно-ориентированных сетях.

Таким образом, система позволит собирать данные о производительности сетевых маршрутов, используемых для доставки нужного контента, и осуществлять обоснованный выбор между ними там, где это возможно, т. е. более эффективно использовать те элементы инфраструктуры, которые находятся вне зоны влияния пользователей сетевых ЭВМ и устройств сети и администраторов корпоративных сетей.

1. Архитектура распределенной системы

Система мониторинга и прогнозирования (СМП), описанная в настоящей работе, имеет распределенную открытую архитектуру и представляет собой совокупность агентов, осуществляющих мониторинг и анализ данных сетевых соединений, выбранных согласно настройкам пользователя. Открытая модульная архитектура агента позволяет использовать различные платформы и инструментальные средства для реализации отдельных модулей, которые взаимодействуют между собой согласно определенным в рамках архитектуры методам. Агенты могут размещаться как на пользовательских устройствах, так и на элементах инфраструктуры, например, прокси-серверах, NAT-серверах, web-серверах, элементах структуры облачных хранилищ и т. п.

Каждый агент регистрирует производительность соединений в некотором заданном множестве сетевых маршрутов на уровне точка—точка (протоколы TCP и UDP). Агент проводит анализ и сжатие собранных данных, предоставляет прогноз производительности наблюдаемых сетевых маршрутов по запросу пользователей и/или приложений. Множество наблюдаемых маршрутов определяется пользователями и/или администраторами сетей и может включать в себя, например, наиболее популярные, имеющие критические требования к производительности и надежности маршруты или маршруты, свойства которых типичны и в силу этого характеризуют группу направлений. Система мониторинга и прогнозирования проводит анализ так называемых "длинных" соединений, так как именно они позволяют получить полную характеристику свойств сетевого маршрута. Заметим также, что если "длинные" соединения демонстрируют удовлетворительную производительность, то и "короткие" будут работать на приемлемом уровне. Обратное неверно. Кроме того, снижение производительности "коротких" соединений часто может быть незаметно пользователям.

Общая архитектура отдельного агента представлена на рис. 1.

Сетевые компоненты являются объектом мониторинга, с них снимают данные для дальнейшей обработки. Это могут быть элементы программного

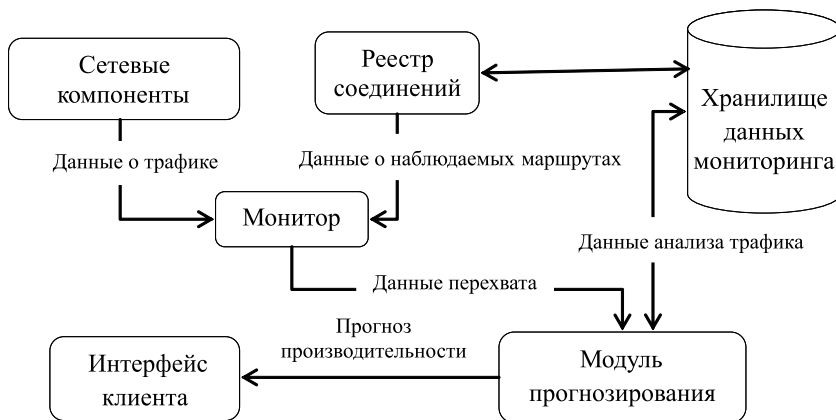


Рис. 1. Высокоуровневая архитектура агента

обеспечения, например, сетевой компонент операционной системы или элементы аппаратного обеспечения, например, сетевая карта в режиме *promiscuous mode*. Реестр соединений имеет каждый агент. Реестр представляет собой список пар маркеров источников и получателей данных, потоки данных между которыми следует регистрировать. Маркерами могут быть IP-адреса источников и получателей, неполные маски IP-адресов, обозначающие группу сетевых ЭВМ, пары IP:порт (протоколы TCP и UDP), а также условные обозначения, введенные пользователем или администратором сети. Последнее связано с наличием во многих сетях механизмов трансляции подлинных IP-адресов. Наличие реестра снижает нагрузку на модуль мониторинга и накладные расходы, позволяя монитору игнорировать трафик потоков, не включенных в реестр. Реестр допускает использование неполных пар, содержащих IP-адрес только источника или только получателя данных. В этом случае монитор будет регистрировать все соединения с указанной конечной точкой, проходящие через сетевую ЭВМ, на которой он установлен. Неполные пары могут использоваться отдельными пользователями, устройства которых всегда являются конечной точкой сетевого маршрута, или прокси-серверами, которые разбивают соединения на два, одно из которых не выходит за пределы локальной сети. Пара, характеризующая сетевой маршрут, также может быть дополнена сведениями о его стоимости. Модуль мониторинга регистрирует данные о соединениях реестра, модуль прогнозирования проводит сжатие и анализ данных мониторинга, а также построение прогнозов производительности сетевых маршрутов по запросам пользователей или сетевых приложений. Взаимодействие с последними осуществляет модуль Интерфейс.

Определяют два уровня детализации в работе агентов. Такое деление обусловлено необходимостью невмешательства в работу сетевой ЭВМ и сокращения накладных расходов, а также объемом сведений о трафике, доступных устройству. Поэтому если сетевая ЭВМ или мобильное устройство не обладают ресурсами, позволяющими вести полноценный

мониторинг, настройки системы позволяют минимизировать объем перехватываемого трафика и аналитический аппарат. Настройки управляют работой двух основных модулей — модуля мониторинга и модуля прогнозирования. Рассмотрим оба уровня детализации более подробно.

1.1. Первый уровень детализации в работе агента

В этом режиме агент размещается на сетевых ЭВМ с высокой производительностью и регистрирует полный набор сведений о поведении соединений на уровне точка—точка. Монитор оперирует на уровне ядра операционной

системы (ОС) и регистрирует следующие сведения:

- об используемой версии протокола TCP и ее опциях (например, опция SACK — *selective acknowledgement*);
- об эволюции размера скользящего окна *cwnd*;
- об ограничении размера скользящего окна, согласованного с получателем *rwin* (*receiver advertised window*);
- о последовательности номеров сегментов, данных и соответствующих временных меток;
- о Maximum Segment Size (MSS, размер сегмента, определяемый сетью);
- о выборках времени кругового оборота;
- о потерях сегментов;
- о мгновенной пропускной способности соединения.

Монитор регистрирует также общие сведения: время начала и завершения соединения; объем переданных данных; число переданных сегментов; среднюю пропускную способность. Примером такого монитора является система *gettcp* [5, 6]. Если наблюдаемое соединение использует протокол UDP, то регистрируются: время кругового оборота; данные о потерях сегментов; данные о средней и мгновенной пропускной способности, а также общие сведения о соединении, описанные выше.

Если монитору недоступно пространство ядра ОС, на первом уровне детализации используются данные перехвата с сетевой карты. Наиболее известным и широко используемым для этой цели инструментарием является библиотека *libpcap* и связанные с ней надстройки *tcpdump* [7] и *Wireshark/tshark*. Заметим, что открытая архитектура агентов и системы в целом позволяет обеспечить реализацию ее модулей на различных платформах и допускает как использование существующих средств и утилит, так и разработку оригинальных подсистем. Монитору, привязанному к сетевой карте, недоступен ряд важных переменных, например, *cwnd* и *rwin*, либо он вносит погрешность в наблюдения. В частности, как было показано в работе [6], наличие механизма TSO в последних версиях TCP приводит к неверной оценке MSS, регистрируемой *tcpdump*. Тем не менее, используя такие мониторы можно зарегистрировать сведения о некоторых из используемых TCP опций,

последовательность номеров сегментов данных и их содержание. Анализ перехваченных пакетов/сегментов во многих случаях позволяет получить выборки времени кругового оборота, уровень потерь, выборки мгновенной пропускной способности, а также общие сведения о соединении, указанные выше. Заметим, что последовательность размеров скользящего окна, и значение *rwin* также могут быть частично восстановлены по номерам сегментов данных и соответствующих подтверждений, однако, как было показано в работе [8], не для всех управляющих алгоритмов TCP. Например, при переходе из фазы Congestion Avoidance в фазу Fast Retransmitt данные, восстановленные таким образом, не являются достоверными. Для UDP-трафика перехват на сетевой карте позволяет получить практически те же сведения, которые доступны в ядре ОС, в силу того, что UDP не использует сложные алгоритмы контроля потока данных.

Зарегистрированные монитором данные о трафике обрабатываются модулем прогнозирования в несколько этапов. Первый этап — сжатие данных. В подавляющем большинстве случаев хранение полных данных перехвата и выборки случайных величин нецелесообразно, поэтому проводится статистическая обработка наблюдений. В частности, по выборкам размера скользящего окна, времени кругового оборота, мгновенной пропускной способности вычисляются оценки: математического ожидания; медианы; вероятности потери сегмента; несмещенная оценка дисперсии. Также могут быть построены квантили и эмпирическая функция распределения, если это указано пользователем в настройках системы. Эти данные и общие сведения о соединении регистрируются в элементе хранилища с указанием на запись в реестре. После этого полные выборки уничтожаются, если пользователем не указано обратное. Заметим, что полный набор перечисленных выше данных об одном TCP-соединении может занимать объем порядка нескольких сотен Мбайт. При этом соответствующая соединению запись в хранилище после обработки будет занимать менее 1 Кбайта.

Второй этап обработки — анализ. Работа алгоритмов протокола TCP часто приводит к росту амплитуды колебаний мгновенной пропускной способности, что, в свою очередь, может привести к сбоям в механизме буферизации потокового видео на стороне получателя. Однако, как показывают последние исследования, несмотря на это, использование протокола TCP как транспортного уровня приложениями передачи видеоданных в режиме реального времени растет в ущерб протоколу UDP, в том числе и в трафике крупных поставщиков контента, таких как порталы федеральных телевизионных каналов или Netflix. Для анализа TCP-потоков на этом этапе СМП использует методы математического моделирования, опирающиеся на результаты исследований производительности протокола TCP. В настоящее время известны аналитические модели версий Reno, NewReno [9] и CUBIC [10]. Наиболее полно исследованной в этом смысле является версия NewReno,

которая до настоящего времени используется многими ОС. Для этой версии построен ряд оценок математического ожидания средней пропускной способности и размера скользящего окна [11, 12], а также точный численный алгоритм линейной сложности для расчета стационарного распределения размера скользящего окна под управлением алгоритма AIMD и пары алгоритмов AIMD и Slow Start [13, 14]. Параметрами этих моделей являются следующие величины: вероятность потери сегмента; размер окна, согласованного с получателем *rwin*; математическое ожидание и/или распределение времени кругового оборота. Таким образом, данные, полученные на этапе сжатия, позволяют получить оценки параметров соответствующих моделей. В свою очередь, такие оценки позволяют рассчитать распределения, представляющие исчерпывающую характеристику свойств сетевого маршрута.

В том случае, если для используемой версии протокола TCP аналитические модели до настоящего времени не разработаны, а также если используется протокол UDP, для анализа применяют статистические методы. Предлагается использовать три основных величины: оценка вероятности потери сегмента (пакета); средняя пропускная способность; среднее время кругового оборота, а также ряд дополнительных величин, которые рассматриваются, если это указано пользователем, и вычисляются с помощью построенных модулем сжатия данных вариационных рядов времени кругового оборота и мгновенной пропускной способности. В частности, такими дополнительными величинами могут быть дисперсия, квантили времени кругового оборота и мгновенной пропускной способности.

Отдельную задачу представляет собой оценка метрики Path Delay Variation (PDV) [15], которую также в некоторых публикациях называют "джиттер". Эта величина представляет собой разность времени прохождения двух сегментов или пакетов данных по сетевому маршруту в одном направлении на стороне получателя. Метрика существенно влияет на работу механизма буферизации потокового видео и, следовательно, на качество видеоряда, наблюдаемого пользователем. Как видно из определения, для получения точных значений PDV требуются данные с обеих конечных точек маршрута. Однако даже в этом случае неверная синхронизация часов и системные факторы могут внести ошибку в измерение PDV. Так, например, монитор *iperf* [16], явно осуществляющий измерение PDV и разработанный именно для анализа TCP- и UDP-потоков, располагается на ЭВМ отправителя и получателя одновременно. В связи с этим предлагается использовать другие оценки, которые также позволяют характеризовать вариabельность сетевого маршрута. Если модуль мониторинга может быть задействован на ЭВМ получателя, то для оценки вариabельности TCP- и UDP-потоков он регистрирует интервалы между поступлениями последовательных сегментов или пакетов данных. Заметим, что в отличие от PDV все такие разности

будут строго положительными за исключением случаев изменения порядка следования пакетов. Если модуль мониторинга работает на стороне отправителя, то будет использована оценка среднего PDV в предположении, что сетевой маршрут является симметричным как $(rtt_i - rtt_{i+1})/2$, где $rtt_i/2$ — оценка односторонней задержки на маршруте.

Результаты анализа и сжатия данных записываются в хранилище. На их основе прогноз производительности сетевого маршрута генерируется по запросу пользователя или сетевого приложения. Взаимодействие при этом осуществляется через интерфейсный модуль. Если получен запрос о метриках, которые могут быть найдены с помощью аналитических моделей, например, мгновенная или средняя пропускная способность или размер скользящего окна протокола TCP, проводят расчеты с использованием упомянутых выше формул и численных методов. Если аналитические и численные методы недоступны, то для построения прогноза используют геометрическое среднее или скользящее среднее. Запросы от заинтересованных сторон поступают в виде набора "конечные точки маршрута, временной интервал" и "параметр, метрика". Например:

- вероятность потери сегмента, доверительный интервал;
- пропускная способность, среднее;
- пропускная способность, минимальное значение;
- RTT, среднее квадратическое отклонение, где RTT — Round Trip Time — время кругового оборота.

Заметим, что данные наблюдений также могут использоваться для исследования наличия в сетях неизвестных пользователям административных воздействий и установления соответствия реализаций протоколов заявленной версии и требованиям Internet Engineering Task Force (IETF).

1.2. Второй уровень детализации в работе агента

Этот режим предназначен для устройств, которые не обладают вычислительными ресурсами и чувствительны к абсолютным объемам накладных расходов. Такие устройства, как правило, являются получателями данных, имеют ограниченные ресурсы, например, устройства пользователей, включая мобильные устройства, или устройства с низкой производительностью, такие как устройства систем IoT, сенсоры и т. д. В связи с ограничениями доступной мощности и, в некоторых случаях, безопасности, мониторинг на уровне ядра ОС для таких устройств невозможен или нежелателен.

В этих случаях модуль мониторинга осуществляет либо минимальный объем пассивного мониторинга в пространстве пользователя, либо использует активное тестирование маршрута для оценки наиболее значимых метрик производительности. Если пассивный мониторинг возможен, то настройки монитора должны определять приемлемые ограничения и регистрировать малый объем данных, необходимых

для прогнозирования, а именно — время кругового оборота, уровень потерь и среднюю пропускную способность соединения. В целях дальнейшего сокращения объема пассивного мониторинга предлагается использовать подсистему активного тестирования маршрута для регистрации уровня потерь и времени кругового оборота на маршруте. Последняя в рамках открытой архитектуры может быть основана, например, на утилите `ping` или ее аналоге. Средняя пропускная способность по-прежнему регистрируется в пассивном режиме, так как деятельность пользователя осуществляется в любом случае, а регистрировать время начала и окончания соединений и объем переданных данных можно элементарными средствами ОС, тогда как идентификация потерянных данных и измерение времени кругового оборота требуют привлечения специальных средств.

Так как ограниченный режим мониторинга не позволяет достоверно установить используемую версию протокола TCP, для аналитического прогнозирования могут использоваться элементарные оценки средней пропускной способности версии NewReno [12, 13]. Иные версии, получившие наибольшее распространение, теоретически демонстрируют более высокую производительность. По этой причине средняя пропускная способность алгоритма AIMD может служить оценкой снизу производительности сетевого маршрута под управлением некоторых других версий. Заметим также, что пропускная способность протокола TCP является монотонно убывающей функцией вероятности потери сегмента и времени кругового оборота. Поэтому тривиальные аналитические модели могут использоваться для сравнения сетевых маршрутов.

Основным методом прогнозирования в условиях ограничений становится статистический анализ. Прогноз может быть дан для оценок средней пропускной способности, среднего времени кругового оборота и вероятности потерь для протоколов TCP и UDP. В качестве методов прогнозирования здесь также используются геометрическое среднее и скользящее среднее. Заметим также, что пропускная способность протокола UDP (без учета потерь данных) в случаях экстремальных ограничений вычислительной мощности может использоваться как оценка сверху пропускной способности TCP, при условии правильной настройки основных параметров UDP.

1.3. Взаимодействие между агентами

Взаимодействие между агентами осуществляется в целях обмена данными. Если отдельный агент не имеет сведений, необходимых для прогноза, то он может переадресовать запрос другому агенту, который предположительно обладает такими данными. Будем рассматривать два возможных случая: в реестре агента нет соединения, о котором получен запрос, или в хранилище агента нет данных требуемого уровня детализации для соединения, о котором получен запрос. Запрос направляется в виде двух пар: "конечные точки маршрута, временной интервал" и

"параметр, метрика" (см. п. 1.1). Если ответ получен, то он перенаправляется пользователю через модуль интерфейса, и недостающее соединение вносится в реестр. Уровень детализации при этом не изменяется.

При определении направления запроса агент следует общей иерархии сети. В первую очередь, запрос направляется конечной точке сетевого маршрута, т. е. агент отправителя направляет запрос получателю и наоборот. Если ответ не был получен (нет действующих агента или агент не располагает нужными сведениями), то запрос направляется на ближайшие в смысле сетевого расстояния элементы инфраструктуры и затем, в случае неудачи, на промежуточные узлы сетевого маршрута. При этом в первой паре начальная точка сетевого маршрута заменяется на IP-адрес ЭВМ агента, которому направляется запрос. Таким образом, ответ будет получен о части сетевого маршрута, лежащей за пределами корпоративной сети или сети поставщика услуг.

Каждый агент поддерживает актуальный список известных ему агентов СМП, содержащий список IP-адресов сетевых ЭВМ, на которых работают эти агенты. При первом запуске агента он опрашивает прокси-сервер, ближайшие маршрутизаторы, сервер поставщика услуг и т. п. о наличии у них агента системы. Если ответ получен, то IP-адреса ЭВМ с обнаруженными агентами заносятся в список и в дальнейшем используются для генерации запроса. Одновременно с этим агент рассылает информационное сообщение о своем присутствии по локальной сети, агенты, получившие сообщение, отправляют ответ, содержащий сведения об их статусе и IP-адрес. Агенты обмениваются также списками известных агентов СМП.

2. Примеры использования

В настоящем разделе описаны возможные сценарии использования СМП в различных контекстах, в частности, сетевыми ЭВМ отдельных пользователей, элементами сетевой инфраструктуры. Приведен также экспериментальный пример решения задачи о выборе поставщика видеоконтента. В разд. 2.3 представлены возможные способы интеграции СМП в парадигму контентоориентированных сетей и схема ее взаимодействия с протоколом Multi-Path TCP.

2.1. Примеры использования распределенной системы мониторинга в рамках существующей инфраструктуры

2.1.1. Сетевые ЭВМ отдельных пользователей

Основная область применения рассматриваемой системы пользователями сети и соответствующими приложениями — оптимизация сетевой активности и построение обоснованных, более эффективных стратегий сетевой активности на основе данных системы. Генерируемый системой прогноз может использоваться для следующих целей или их комбинации:

- выбор источника контента;

- выбор времени обращения к источнику контента;

- выбор сетевого маршрута там, где это возможно.

Если один и тот же контент или контент одного класса предоставляется различными источниками, то генерируемый системой прогноз производительности соответствующих сетевых маршрутов позволяет пользователю или сетевому приложению сделать обоснованный выбор наиболее эффективного источника. Мониторинг позволит также выделить периоды наименьшей загрузки ресурса, когда обращение к нему наиболее выгодно. В качестве примеров можно привести ресурсы, предоставляющие видео- и аудиоконтент, новостные, образовательные ресурсы и удаленные игровые ресурсы.

Если сеть, предоставляющая доступ к сетевым услугам, позволяет настраивать сетевые маршруты внутри фрагмента сети, например, внутри корпоративной сети, то сетевая ЭВМ может изменять настройки шлюза, выбирать маршрутизатор, выбирать прокси-сервер или не использовать прокси-сервер вообще, а также выбирать между доступными Wi-Fi сетями, то во всех этих случаях после того как накоплена история наблюдений, прогноз производительности позволит выделить наиболее эффективный сетевой маршрут.

2.1.2. Элементы инфраструктуры корпоративных сетей

Элементы инфраструктуры корпоративных сетей, такие как NAT и прокси-сервера, так называемые "внешние" маршрутизаторы, непосредственно связанные с поставщиками услуг, имеют доступ к значительным объемам данных о трафике и обладают большими вычислительными ресурсами. Использование системы мониторинга и прогнозирования в условиях наличия доступа к этим данным позволит решать следующие задачи:

- получать сведения о производительности наиболее часто используемых в организации сетевых маршрутов;
- предоставлять сотрудникам и клиентам рекомендации по оптимизации их сетевой активности;
- оценивать эффективность топологии и конфигурации корпоративной сети при обеспечении доступа к внешним ресурсам.

Заметим, что наличие централизованной системы мониторинга позволит снять нагрузку с пользовательских устройств.

Распределенная система мониторинга может также использоваться поставщиками контента, такими как крупные образовательные порталы, видео- и аудиохостинги. Для таких организаций объектами мониторинга могут быть сетевые маршруты, для которых источник — один из серверов поставщика, и получатель — элемент географически локализованного семейства IP-адресов. Выявление наименее производительных направлений позволит скорректировать свойства сетевой инфраструктуры, если это возможно, либо изменить маркетинговую по-

литику в регионе, адаптируя ее к текущему уровню производительности. Заметим также, что сетевая инфраструктура развивается иногда очень активно, поэтому непрерывный анализ и мониторинг дают доступ к актуальной информации и позволяют вносить своевременные воздействия, управляющие этими процессами.

2.2. Экспериментальный пример анализа

Рассмотрим задачу о выборе поставщика видеоконтента. В настоящем разделе приведены экспериментальные данные и их анализ в рамках архитектуры, описанной выше. В экспериментах рассматривали два федеральных (Первый канал и Вести.ru) и один региональный телевизионный канал (ВГТРК "Карелия") и три альтернативных поставщика одного и того же контента: web-портал соответствующего телеканала, хостинг youtube и страницу канала в социальной сети "ВКонтакте". В качестве модулей мониторинга и подсистемы сжатия данных модуля прогнозирования использовали утилиты `tcpdump` и `tcptrace` [17] соответственно. В первой серии экспериментов получатель данных был расположен в корпоративной сети Петрозаводского государственного университета, имеющей внешний канал "Ростелеком", во второй серии доступ был предоставлен через выделенный канал регионального поставщика услуг "Сампо". Заметим, что все источники и поставщики для передачи видеоряда использовали транспортный протокол TCP, объем данных, переданных по протоколу UDP оказался пренебрежимо мал (менее 1 Мбайта).

Утилита `tcpdump` фиксировала: время регистрации TCP-сегмента; тип (данные или подтверждение `ack`) и номер в последовательности; время начала соединения; время его завершения; объем сегмента в байтах; сведения об источнике и отправителе — крайних точках сетевого маршрута. Данные, содержащиеся в самих TCP-сегментах, не извлекали и не

рассматривали. Заметим, что даже в условиях подробной регистрации на пользовательском уровне, без обращения к уровню ядра, этап сжатия данных весьма важен. Так, например, регистрационный файл `tcpdump` о передаче 67,3 Мбайт данных в развернутом виде имеет собственный размер 58 Мбайт. Размер тех же данных после сжатия составляет несколько Кбайт. На этапе сжатия выделяли отдельные TCP-соединения, о каждом из которых формировалась отдельная запись, содержащая сведения: о конечных точках маршрута; об общем объеме данных, переданных в обоих и отдельно в каждом направлении, выраженные в байтах и сегментах; о длительности соединения; о времени его начала; о пропускной способности; о размере MSS. К сожалению, точное определение RTT, числа потерянных сегментов и значения `rwin` на стороне получателя затруднено, и данные, предоставляемые утилитой `tcptrace`, не являются вполне достоверными. Поэтому основной метрикой для прогнозирования и рекомендаций была выбрана пропускная способность (ПС). Рассматривали только соединения объемом более 1 Мбайта. Результаты анализа данных после сжатия приведены в табл. 1.

С учетом представленных в табл. 1 данных, наилучшую среднюю пропускную способность демонстрирует сетевой маршрут, в котором источником данных является web-портал Первого канала. Однако он же, обладая наибольшей вариабельностью, и демонстрировал в экспериментах наибольший размах пропускной способности. Web-портал `vesti.ru` демонстрирует существенно худшую среднюю производительность при сравнимой относительной вариабельности. Наиболее стабильным является сетевой маршрут к web-порталу ВГТРК Карелия, для которого среднее квадратическое отклонение составляет 12 %. Тот же контент, полученный с видеохостинга youtube, демонстрирует сравнимые средние значения, но существенно большую вариабельность. Страницы

Таблица 1

Первая серия экспериментов

Источник	Средняя ПС, Мбит/с	Среднее квадратическое отклонение ПС, Мбит/с	Максимальная ПС, Мбит/с	Минимальная ПС, Мбит/с	Медиана ПС, Мбит/с
Первый канал (web-портал), полноэкранное видео	3 112 14,54	1 642 836,15	6 136 464,49	892 358,30	3 195 15,06
Первый канал (web-портал)	2 231 34,03	1 116 581,09	5 505 990,21	1 286 399,02	1 872 788,23
Первый канал (ВКонтакте)	2 940 00,01	1 291 776,76	5 494 456,36	1 292 428,27	3 126 287,56
Vesti.ru (web-портал)	1 053 27,22	451 113,63	1 179 082,20	562 149,85	1 020 542,90
Vesti.ru (ВКонтакте)	2 223 40,67	1 317 357,13	4 346 156,44	1 006 758,64	1 772 283,72
ВГРТК Карелия (web-портал)	1 864 94,31	231 708,46	2 403 226,94	1 595 644,03	1 849 410,91
ВГРТК Карелия (youtube)	1 729 09,25	701 358,41	3 515 431,56	922 816,32	1 872 008,81

Таблица 2

Вторая серия экспериментов

Источник	Файл 1, Мбит/с	Файл 2, Мбит/с
Первый канал (web-портал), полноэкранное видео	10,7	2,9
Первый канал (web-портал)	1,6	3,2
Первый канал (youtube), полноэкранное видео	2,6	0,87
Первый канал (youtube)	1,7	2,3
Первый канал (Вконтакте)	0,89	0,91

производителей контента в сети "Вконтакте" в подавляющем большинстве случаев демонстрируют более низкую производительность, чем оригинальные порталы, хотя анализ IP-адресов показывает, что сам видеоконтент передается с тех же сетевых ЭВМ. Таким образом, на исследуемом сетевом маршруте web-портал Первого канала может поставлять видеоряд качеством 720p H.264 или телевидение стандартного разрешения. Все остальные источники обеспечивают качество ниже 720p.

Во второй серии экспериментов пользовательской ЭВМ была получена серия видеоданных, содержащих

новостные передачи Первого канала. Получателем являлась пользовательская ЭВМ. Доступ к сети предоставлялся региональным поставщиком услуг "Сампо". Из каждого из трех источников были получены два видеоролика длительностью около 2 мин (табл. 2).

Эти данные не противоречат качественным оценкам, описанным выше.

В целях дополнительного анализа также были получены оценки времени кругового оборота и уровня потерь данных на сетевых маршрутах с источниками домена misp.ru (новости Первого канала) и tv-karelia.ru. Для этих данных были проведены расчеты пропускной способности по формуле (1), полученной в работе [11], и с использованием алгоритма расчета распределения скользящего окна, предложенного в работе [13], который позволяет учитывать параметр $rwin$ алгоритма AIMD протокола TCP:

$$T = \frac{MSS}{RTT} \sqrt{\frac{3}{2p}}, \quad (1)$$

где RTT — среднее время кругового оборота, p — вероятность потери сегмента.

Использовалась оценка $rwin$, полученная утилитой tcptrace. Результаты расчетов приведены в табл. 3.

Как следует из данных табл. 3, на маршруте к ресурсу tv-karelia.ru протокол TCP теоретически может

Таблица 3

Результаты моделирования

Источник данных	Среднее RTT, мс	Вероятность потери сегмента	Оценка $rwin$ tcptrace, сегменты	Средний размер скользящего окна [11]/[13], сегменты	Пропускная способность [11]/[13]/Эксп, Мбит/с
*.misp.ru	30,7	0,0032	22,3	28,59/17,7	8,2/6,7/6,1
tv-karelia.ru	39,3	<0,001	20,7	46,77/19,4	13,5/5,8/1,9

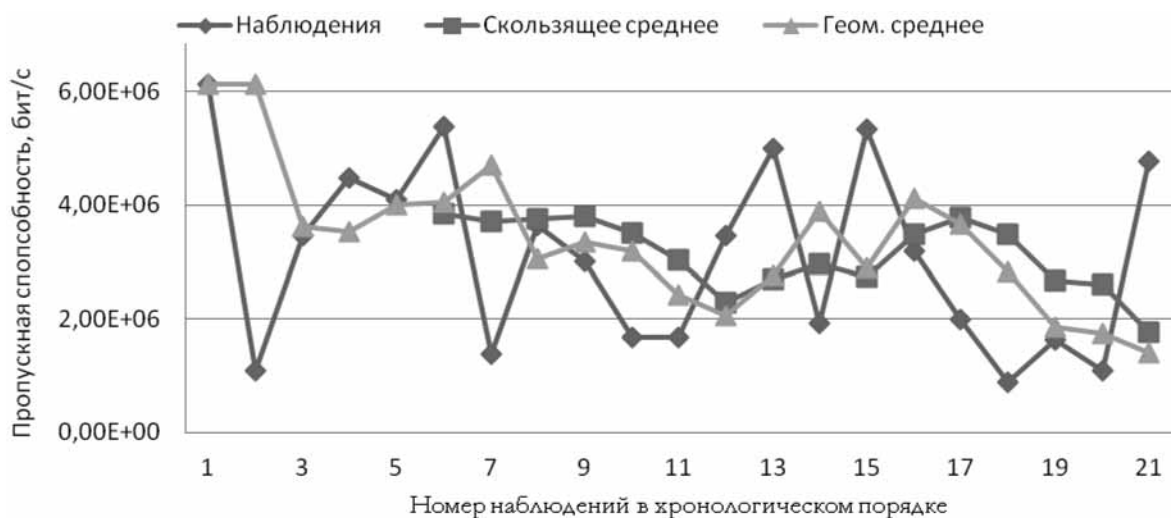


Рис. 2. Прогноз временного ряда. Первый канал



Рис. 3. Прогноз временного ряда. ВГТРК Карелия

показывать существенно более высокую производительность. Результаты моделирования свидетельствуют о наличии дополнительных ограничений пропускной способности, применяемой администрацией ресурса.

Теперь перейдем к задаче прогнозирования производительности. Модуль прогнозирования на основе данных наблюдений использует два основных метода: скользящее среднее и геометрическое среднее. В условиях, когда значения среднего квадратического отклонения близки к 50 % математического ожидания, погрешность прогнозов, основанных на этих методах, также является значительной. Тем не менее, даже в этих условиях прогноз позволяет сравнить потенциальные источники данных и дает оценку производительности сетевого маршрута, которая может улучшить стратегию пользователя. На рис. 2, 3 представлены прогнозы производительности сетевых маршрутов, в которых источниками данных являются web-порталы Первого канала и ВГТРК Карелия с параметрами 0,5 для геометрического среднего и 5 наблюдений для скользящего среднего.

2.3. Использование распределенной системы мониторинга и прогнозирования в рамках перспективных сетевых технологий

2.3.1. Контентно-ориентированные сети

В настоящее время разработано несколько родственных парадигм сетевой архитектуры, предполагающих фундаментальное изменение основных принципов маршрутизации и коммутации пакетов [1]. Эти парадигмы имеют общее ядро, состоящее в том, что основой адресации и маршрутизации в сетях должны стать не сетевые ЭВМ, а содержательные данные. Пользователь должен запрашивать не соединение с сетевой ЭВМ, а некоторые данные, источник которых будет определяться элементами

инфраструктуры, а затем доставляться пользователю наиболее эффективным образом. Эти парадигмы предполагают дублирование данных в географически локализованных хранилищах и за счет сетевого кэширования. Их основная цель — повысить производительность и качество услуг в сети.

Важным компонентом инфраструктуры этих архитектур является подсистема управления контентом, которая получает запросы о доставке данных, идентифицирует потенциальные источники и перенаправляет запрос одному из них. К настоящему времени все алгоритмы этих систем построены в предположении, что наилучшим в смысле производительности источником данных является географически ближайший источник или источник с наименьшим сетевым расстоянием до получателя. Это предположение не всегда оправдано. Также очевидно, что интенсивное использование таких источников приведет к перегрузке соответствующих маршрутизаторов и каналов связи. Использование средств мониторинга и прогнозирования как элемента подсистемы управления контентом позволит контролировать производительность сетевых маршрутов, осуществлять обоснованный выбор источника и балансировать нагрузку на хранилища данных.

2.3.2. Протокол MPTCP

Протокол Multi Path Transmission Control Protocol (MPTCP) [18] является модернизированной версией протокола TCP и предназначен для более эффективного использования сетевой инфраструктуры и балансирования нагрузки ее элементов. Протокол способен открывать TCP-соединение по нескольким каналам связи одновременно в целях увеличения общей пропускной способности и/или надежности соединения. Протокол сохраняет свойство TCP-friendly [11, 12], т. е. его деятельность не должна дискриминировать

соединения под управлением стандартного TCP-протокола. С этой целью общий поток, генерируемый протоколом, разделяется по каналам связи пропорционально их мощности. Однако в силу того, что источник данных не обладает сведениями о мощности используемых каналов, соотношение устанавливается в течение соединения с помощью модифицированных версий алгоритмов Slow Start и AIMD. Система мониторинга и прогнозирования, обладая сведениями о потенциальной производительности каналов связи, может использоваться MPTCP для начального балансирования потоков по следующей схеме.

- Инициатор соединения отправляет пакет SYN с флагом MP_CAPABLE, получив SYN/ACK, направляет запрос агенту СМП о математических ожиданиях размера скользящего окна $cwnd_1$ и rtt_1 на сетевом маршруте. При этом математическое ожидание размера скользящего окна рассчитывается с использованием алгоритма, описанного в работе [13].

- Получив пакет MP_JOIN, инициатор также направляет агенту СМП аналогичный запрос о сетевом маршруте для подпотока и получает сведения $cwnd_2$, rtt_2 .

- Протокол MPTCP определяет соотношение пропускных способностей подпотоков как $(cwnd_1/rtt_1)^* (rtt_2/cwnd_2)$.

Заключение

Описана система мониторинга и прогнозирования производительности сетей передачи данных. Система представляет собой распределенное множество агентов, осуществляющих мониторинг производительности соединений на уровне точка—точка, регистрацию данных мониторинга и предоставление сведений о предполагаемой производительности сетевого маршрута по запросам сетевых протоколов, приложений, сетевых устройств. Предлагаемая система может применяться для повышения эффективности использования сетевых ресурсов, оптимизации работы протоколов, балансирования нагрузки. Приведены возможные сценарии использования системы, а также пример экспериментального анализа для задачи о выборе источника видеоданных.

Дальнейшими направлениями работы по развитию системы предполагаются разработка методов автоматического определения перспективных направлений мониторинга, совершенствование методов прогнозирования, разработка системы интеграции и анализа данных отдельных агентов в рамках концепции [19].

Список литературы

1. Carzanigo A., Papalini M., Wolf A. L. Content-Based Publish/Subscribe Networking and Information-Centric Networking // ICN'11, August 19, 2011, Toronto, Ontario, Canada. 2011. P. 56—61.

2. Fundamental Limitations of current Internet and the path to Future Internet. EC FIArch Group2 Release Date: 1 March 2011.

3. Border J., Kojo M., Griner J., Montenegro G., Shelby Z. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, RFC 3135, June 2001. URL: <https://tools.ietf.org/html/rfc3135>

4. Open Flow. Finding Feature Information. Техническая документация. URL: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960xr/software/15_2_6_e/configuration_guide/b_1526e_consolidated_2960xr_cg/openflow.pdf

5. Санников А. А., Богоявленская О. Ю., Богоявленский Ю. А. Система анализа поведения и мониторинга производительности соединений транспортного уровня Интернет // Программная инженерия. 2013. № 12. С. 27—33.

6. Sannikov A. A. Bogoiavlenskaia O. I., Bogoyavlenski I. A. GetTCP+: Performance monitoring system at transport layer // Proceedings of NEW2AN/ruSMART. 2013. LNCS. Vol. 8121. P. 236—246.

7. tcpdump/Libpcap public repository. URL: <http://www.tcpdump.org>

8. Пономарев В. А., Богоявленская О. Ю., Богоявленский Ю. А. Проверка адекватности и модификация модели случайного потока, генерируемого транспортным протоколом TCP в сети передачи данных // Труды международного семинара "Распределенные компьютерные и телекоммуникационные сети: теория и приложения (DCCN 2007)". 2007. Том 2. С. 59—64.

9. Allman M., Paxson V. TCP Congestion Control, RFC 5681, 2009. URL: <https://tools.ietf.org/html/rfc5681>

10. Ha S., Rhee I., Xu L. CUBIC: a new TCP-friendly high-speed TCP variant // Newtler ACM SIGOPS Operating Systems. 2008. Vol. 42 (5). P. 64—74.

11. Floyd S., Fall F. Promoting the use of end-to-end congestion control in the Internet // IEEE/ACM Transactions on Networking. 1999. Vol. 7, No. 4. P. 458—472.

12. Padhey J., Firoiu V., Towsley D., Kurose J. Modeling TCP Throughput: A Simple Model and its Empirical Validation // IEEE / ACM Transactions on Networking. 2000. Vol. 8, No. 2. P. 133—145.

13. Богоявленская О. Ю. Анализ случайного потока, генерируемого транспортным протоколом с обратной связью, в сети передачи данных // Автоматика и телемеханика. 2003. № 12. С. 60—68.

14. Богоявленская О. Ю. Вероятностная модель алгоритмов протокола распределенного управления сети Интернет // Автоматика и телемеханика. 2009. № 1. С. 119—129.

15. Claise B. Packet Delay Variation Applicability Statement. RFC 5481. 2009. URL: <https://tools.ietf.org/html/rfc5481>

16. Iperf-TCP / UDP Bandwidth Measurement tool. URL: <http://iperf.fr/>

17. Tcptrace Homepage. URL: <http://www.tcptrace.org/>

18. Ford A., Raiciu C., Handley M., Bonaventure O. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824b 2107. URL: <https://tools.ietf.org/html/draft-ietf-mptcp-rfc6824bis-09>

19. Макаров В. Л., Бахтизин А. Р., Сушков Е. Д. и др. Агент-ориентированные модели: мировой опыт, технические возможности реализации на суперкомпьютерах // Вестник Российской Академии наук. 2016. Том 86, № 3. С. 252—262.

The Distributed Multi-agent System for Monitoring and Forecasting of the Networking Transport Layer Performance

O. V. Bogoiavlenskaia, e-mail: olbgvl@cs.karelia.ru, Petrozavodsk State University, Petrozavodsk, 185910, Russian Federation

Corresponding author:

Bogoiavlenskaia Olga V., Associate Professor, Petrozavodsk State University, Petrozavodsk, 185910, Russian Federation, E-mail: olbgvl@cs.karelia.ru

Received on October 22, 2017
Accepted on November 15, 2017

The paper describes the architecture of the distributed multi-agent system for monitoring and forecasting of the networking transport layer performance. The system consists of the set of the agents which monitor, analyze captured data and forecast the performance of most intensively used end-to-end paths. The forecast allows users and network application to choose most efficient strategies to utilize the capacity of network resources. Therefore the end-users, mobile devices, LAN administrations, and network applications can choose content provider, end-to-end-path or data source to avoid drawbacks of congested and/or unreliable end-to-end paths. Content providers can use the forecast to monitor QoS and/or QoE of their users and subscribers and thus adjust marketing policies and investment plans. The high level architecture consists of the monitoring module, forecasting module, list of the end-to-end paths under analysis, compressed data store and interface module. We define two levels of detailing for the agents architecture to reduce monitoring and processing overhead. The agents exchange data they possess about end-to-end paths performance and maintain list of agents they can communicate. The set of use cases is presented. The example of the experimental analysis for choosing video-content provider is presented as well.

Keywords: monitoring, Linux kernel, network performance, transport layer, open architecture, distributed systems, probabilistic modeling, time-series analysis, video-streaming, multi-agent systems

For citation:

Bogoiavlenskaia O. V. The Distributed Multi-agent System for Monitoring and Forecasting of the Networking Transport Layer Performance, *Programnaya Ingeneria*, 2018, vol. 9, no. 1, pp. 11–21.

DOI: 10.17587/prin.9.11-21

References

1. Carzanigo A., Papalini M., Wolf A. L. Content-Based Publish/Subscribe Networking and Information-Centric Networking, *ICN'11*, August 19, 2011, Toronto, Ontario, Canada, 2011, pp. 56–61.
2. Fundamental Limitations of current Internet and the path to Future Internet. EC FIArch Group2 Release Date: 1 March 2011.
3. Border J., Kojo M., Griner J., Montenegro G., Shelby Z. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, RFC 3135, June 2001, available at: <https://tools.ietf.org/html/rfc3135>
4. Open Flow. Finding Feature Information. Technical documentation, available at: URL:https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960xr/software/15_2_6_e/configuration_guide/b_1526e_consolidated_2960xr_cg/openflow.pdf
5. Sannikov A. A., Bogoiavlenskaia O. Yu., Bogoiavenskii I. A. Sistema analiza povedeniya i monitoringa proizvoditel'nosti soedinenij transportnogo urovnya Internet (System for Behavior Analysis and Performance Monitoring of Internet Transport Layer), *Programnaya Ingeneria*, 2013, no. 12, pp. 27–33 (in Russian).
6. Sannikov A. A., Bogoiavlenskaia O. I., Bogoyavlenskii I. A. GetTCP+: Performance monitoring system at transport layer, *Proceedings of NEW2AN/ruSMART*, 2013, LNCS, vol. 8121, pp. 236–246.
7. tcpcap/Libpcap public repository. URL:<http://www.tcpdump.org>
8. Ponomarev V. A., Bogoiavlenskaia O. I., Bogoyavlenskii I. A. Proverka adekvatnosti i modifikacija modeli sluchajnogo potoka, generiruемого transportnym protokolom TCP v seti peredachi dannyh (Verification of the adequacy and modification of the model random stream generated by the transport Protocol TCP in data transmission network), *Trudy mezhdunarodnogo seminara "Raspredelennye komp'yuternye i telekommunikacionnye seti: teorija i prilozhenija (DCCN 2007)"*, 2007, vol. 2, pp. 59–64 (in Russian).
9. Allman M., Paxson V. TCP Congestion Control, RFC 5681, 2009, available at: <https://tools.ietf.org/html/rfc5681>
10. Ha S., Rhee I., Xu L. CUBIC: a new TCP-friendly high-speed TCP variant, *Newstler ACM SIGOPS Operating Systems*, 2008, vol. 42 (5), pp. 64–74.
11. Floyd S., Fall F. Promoting the use of end-to-end congestion control in the Internet, *IEEE / ACM Transactions on Networking*, 1999, vol. 7, no. 4, pp. 458–472.
12. Padhey J., Firoiu V., Towsley D., Kurose J. Modeling TCP Throughput: A Simple Model and its Empirical Validation, *IEEE / ACM Transactions on Networking*, 2000, vol. 8, no. 2, pp. 133–145.
13. Bogoiavlenskaia O. Yu. Analysis of the random flow generated by the feedback transport protocol in a data transfer network, *Automation and Remote Control*, 2003, vol. 64, no. 12, pp. 1882–1889.
14. Bogoiavlenskaia O. Yu. Probabilistic model of the algorithms of distributed control protocol in the internet network, *Automation and Remote Control*, 2009, vol. 70, no. 1, pp. 107–117.
15. Claise B. Packet Delay Variation Applicability Statement. RFC 5481. 2009, available at: <https://tools.ietf.org/html/rfc5481>
16. Iperf-TCP / UDP Bandwidth Measurement tool, available at:<http://iperf.fr/>
17. Tcptrace Homepage, available at: <http://www.tcptrace.org/>
18. Ford A., Raiciu C., Handley M., Bonaventure O. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824b 2107, available at: <https://tools.ietf.org/html/draft-ietf-mptcp-rfc6824bis-09>
19. Makarov V. L., Bakhtizin A. R., Sushko E. D., Vasev V. A., Borisov V. A., Roganov V. A. Supercomputer technologies in social sciences: Agent-oriented demographic models, *Herald of the Russian Academy of Sciences*, 2016, vol. 86, no. 3, pp. 248–257.

С. В. Поршневу, д-р техн. наук, проф., e-mail: s.v.porshnev@urfu.ru,
О. А. Пономарева, ст. преподаватель, e-mail: o.a.ponomareva@urfu.ru,
ФГАУ ВО Уральский федеральный университет, г. Екатеринбург,
А. М. Бородин, канд. техн. наук, разработчик, e-mail: amborodin@urfu.ru, ООО "Яндекс",
г. Екатеринбург, **С. Г. Мирвода**, руководитель проекта, e-mail: s.g.mirvoda@urfu.ru,
ООО "Октоника", г. Екатеринбург

Модификация функции penalty R-дерева над обобщенным деревом поиска индексов для повышения производительности модуля cube PostgreSQL

Обобщенное индексное дерево поиска (*Generalized Index Search Tree — GiST*), применяемое для обработки запросов, значительно упрощает разработку методов доступа к данным. Оно позволяет реализовать управление памятью в обобщенном коде, обеспечивает восстановление хранилищ данных после сбоев и организует эффективную одновременную работу различных пользователей с данными. При использовании *GiST* разработчику метода доступа к хранилищу данных оказывается достаточным реализовать только несколько операций проектируемого алгоритма в соответствии с интерфейсом (*Application programming interface — API*) *GiST*. Обсуждены варианты оптимизации функции *penalty* при реализации R-дерева в рамках фреймворка *GiST PostgreSQL*. Проведен анализ возможных направлений дальнейшего развития API с целью создания API, оптимального для современных алгоритмов поиска в многомерных данных.

Ключевые слова: обобщенное индексное дерево поиска, метод доступа к многомерным данным, *PostgreSQL*, R-дерево

Введение

Современные технологии управления хранилищами данных (ХД) обеспечивают решение задач компактного хранения данных, разрешения конфликтов записи данных и эффективного выполнения запросов на извлечение из ХД данных, удовлетворяющих условию запроса. Запросы выполняются с помощью так называемых методов доступа, объединяющих в себе алгоритмы и структуры данных поиска в СУБД. В современных системах управления реляционными базами основным методом доступа к данным является *B*-дерево [1], позволяющее извлекать данные с помощью первичного ключа и совершения других операций [2]. Обобщенное индексное дерево поиска (*GiST*) — метод доступа, позволяющий реализовать на более высоком уровне абстракции структуру индекса, основанную на использовании сбалансированных деревьев. Использование *GiST* дает возможность разработчику сосредоточиться на конкретных деталях методов доступа, не затрачивая усилий на реализацию дре-

вовидной структуры в рамках ядра базы данных, интеграции языка запросов, поддержке планировщика запросов и т. д.

Впервые *GiST* было предложено Дж. Хеллерштейном в работе [3], дальнейшие исследования проводили М. Корнакер [4]. Позднее *GiST* в приложении к *PostgreSQL* развивали О. Бартунов и Т. Сигаев [5]. Текущая реализация *GiST* в *PostgreSQL* обеспечивает возможность использования различных деревьев для типов данных, имеющих соответствующий класс операторов (*opclass*), которые описывают специфику индекса для конкретного типа данных. Для объявления в *PostgreSQL* типа данных индексируемым *GiST* разработчику *opclass* достаточно описать следующие основные функции:

1) *Split* — функцию разделения набора экземпляров ключей на две части;

2) *Penalty calculation* — функцию определения "штрафа" расширения R-дерева при объединении двух ключей;

3) *Collision check* — функцию, определяющую наличие пересечения двух ключей;

4) *Unification* — функцию, объединяющую два ключа в один так, что объединенный ключ пересекается с обоими входными ключами.

Здесь функции 1, 2 и 4 используются для построения дерева индексов, а функция 3 используется для выполнения запросов к построенному дереву. В общем случае конструкцию дерева можно рассматривать как последовательную вставку набора данных в дерево индексов. Вставка — это рекурсивный алгоритм, выполняемый для каждого узла, начиная с корня дерева. Этот алгоритм ищет узел дерева для записи (иначе говоря, ссылку на данные) с минимальной "стоимостью" добавления элемента. По выбранной ссылке ключ обновляется функцией 4. Если алгоритм вызывается на листовой странице дерева (странице, не имеющей ссылок на другие страницы), то он просто помещает вставляемый элемент. Если узел заполнен, то запускается восходящая последовательность разделений с помощью функции 1. Например, если для функций 1—4 выбрать обычные прямоугольники, будет построено обычное *R*-дерево [6].

В *GiST PostgreSQL* в соответствии каждому узлу обобщенного дерева ставится страница (в терминологии *PostgreSQL* — *block* или *page*). Это позволяет обрабатывать данные, превышающие по размеру объем ОЗУ, за счет размещения некоторых страниц в буфере ОЗУ, а остальных — в постоянной памяти. В алгоритмах *GiST*, реализованных в *PostgreSQL*, не используется рекурсия на стеке, в целях недопущения его переполнения. Также *GiST PostgreSQL* позволяет использовать в одном индексе несколько разных типов данных с различными *opclass*. Анализ результатов использования *GiST* показал возможность увеличения производительности операций вставки/обновления без существенной модификации кода и использования новых алгоритмов за счет улучшения алгоритмов пространственного индексирования [7, 8] и использования внутристраничного индексирования [9].

В данной работе обсуждаются результаты дальнейших исследований, в ходе которых разработана модифицированная функция *penalty* *R*-дерева *PostgreSQL* над обобщенным деревом поиска индексов, обеспечивающая повышение производительности модуля *cube*. Разработанная функция является примером, свидетельствующим о возможности устранения некоторых ограничений, имеющихся в *GiST*. Отметим, что данный подход может быть использован при разработке и других подобных функций.

Алгоритм, реализованный ранее в функции *penalty*

Алгоритм вставки нового индекса *e* в *R*-дерево, реализованный ранее в функции *penalty*, представляет собой модификацию оригинального алгоритма вставки в *R*-дерево, предложенного А. Гутманом в работе [6].

Данный алгоритм реализуется следующей последовательностью действий.

Шаг 1. [Найти позицию для новой записи] Вызвать *choose_leaf*, чтобы выбрать листовой узел *l*, в который нужно поместить *e*.

Шаг 2. [Добавить запись в листовой узел].

а) Если в *l* есть место для другой записи, разместить в *l* запись *e*.

б) Иначе вызвать *split_node* для получения частей *l* и *2*, содержащих *e* и все прежние записи из *l*.

Шаг 3. [Распространять изменения вверх] Вызвать *adjust_tree* для узла *l*, а также узла *2*, если было выполнено разделение.

Шаг 4. [Нарращивание дерева в высоту] Если распространение разделения узлов привело к разделению корня дерева, то создать новый корень, в котором дочерними элементами будут два результирующих узла.

В процессе выбора листового узла *choose_leaf* для записи нового индекса *e* итеративно выбирается поддерево с наименьшим значением функции увеличения объема ограничивающего параллелотопа *bounding box*. При этом учитывается увеличение объема, которое произойдет в результате вставки записи *e* в поддерево *R*-дерева. В случае равных изменений объемов двух поддеревьев выбирается поддерево с *bounding box* наименьшего объема. Рекурсивный поиск поддерева *n* продолжается, пока *n* не окажется конечным узлом.

Большинство из действий, описанных выше, реализованы в рамках *GiST*. Функции для объединения ключей, разделения узлов и функцию расчета минимальной стоимости добавления элемента для реализации *choose_leaf* предоставляет модуль *cube*, являющийся объектом исследования данной работы.

Во время выбора поддерева *GiST* вызывает функцию *penalty* из модуля *cube* и выбирает элемент с минимальным значением функции *penalty*. Таким образом, производительность итогового индекса напрямую зависит от качества реализации функции *penalty*. Отметим, что в оригинальной статье [6] вопрос о расчете "наименьшего расширения" рассмотрен недостаточно подробно. В связи с этим были проведены дальнейшие исследования, которые показали, что при вычислении "расширения" объема можно и целесообразно использовать умножение ребер (объем многомерной фигуры) или сумму длин ребер (периметр многомерной фигуры) [10].

В настоящее время в модуле *cube* расчет объема является единственным методом выбора поддерева для вставки. Поэтому значение *penalty* рассчитывается как разница между возможным размером после вставки элемента в поддерево и текущим размером поддерева. В некоторых базах данных (например, в реализации *MySQL InnoDB*) существует возможность выбора стратегии расширения в момент компиляции (через *#define* между расчетом объема и периметра).

Авторы настоящей работы провели несколько испытаний процедуры вставки для *R*-деревьев с функцией *penalty* на основе объема и периметра. Для испытаний использовались сгенерированные случайные

данные. По их результатам был сделан вывод о том, что производительность функции *penalty* в большом числе проведенных испытаний является вполне приемлемой. Однако в ряде случаев наблюдалась деградация производительности доступа к данным, в том числе при описанных далее обстоятельствах:

1) при индексировании данных меньшей размерности, чем у дерева (например, двумерных данных в трехмерном пространстве или любых данных, плотно расположенных на параллельных осях гиперплоскостях) запросы могут деградировать до полного сканирования индекса (в этой ситуации, поскольку любой объем *R*-дерева равен нулю, функция *penalty* всегда возвращает ноль, после чего индексу присваивается значение случайного числа, генерируемого в *PostgreSQL* генератором псевдослучайных чисел);

2) при индексировании многомерных нормализованных или стандартизованных данных с небольшими отклонениями на каждой оси, так как результат перемножения множества мелких ребер зачастую оказывается близким к нулю (например, 32-битные данные типа *float*, широко используемые в машинном обучении, будучи размещенными в 40-мерном кубе с длиной ребра 0,01 определяют куб с объемом, равным машинному нулю).

Замена функции *penalty* на функцию, основанную на вычислениях периметра, защищена от описанных негативных сценариев. Однако данная замена приводит к снижению эффективности дерева индексов в остальных случаях.

Модификация функции *penalty*

Для устранения описанных выше проблемных вопросов авторы предлагают использовать функцию штрафа, основанную на совместном применении обоих способов оценки "расширения" объема многомерных типов: сначала вычислить объем многомерной фигуры и, в случае его деградации, — периметр. Данная функция, в случае хорошей производительности, будет вычислять значения *penalty* на основе объема, а в случае деградации при расчете объема организует индекс в соответствии с идеями оригинальной статьи А. Гуттмана [6].

Согласно *GiST API*, функция *penalty* должна возвращать неотрицательное 32-битное значение *float* для некоторого заданного узла поддерева и вставляемого узла. При этом нулевое значение функции соответствует наилучшему варианту вставки. Таким образом, чем больше возвращаемое значение, тем менее предпочтительна вставка индекса в данное поддерево. В большинстве случаев 32-битное значение *float* реализовано в соответствии со стандартом IEEE 754.

При использовании предложенного подхода весь доступный диапазон *float* разделяется на перечисленные далее четыре области.

Область 0. Вставка не требует увеличения *bounding box* поддерева, объем равен нулю, поэтому *penalty* вычисляется через периметр.

Область 1. Расширение не требуется, поддерево имеет ненулевой объем, *penalty* вычисляет объем.

Область 2. Увеличение объема равно нулю, но периметр *bounding box* увеличивается, и *penalty* вычисляется как изменение периметра.

Область 3. Увеличение объема, отличное от нуля, *penalty* вычисляет разницу объема *bounding box* после вставки и объема перед вставкой.

При этом, очевидно, обеспечивается выполнение условия, что любое значение *penalty* области *X* больше любого соответствующего значения области *X-1*. Таким образом, обеспечивается применение логики *R*-дерева по выбору поддерева в тех случаях, когда это не приводит к сравнению нулевых объемов, а в случае сравнения нулевых объемов алгоритм переходит к сравнению периметров.

Битовый сдвиг в IEEE 754 floats

В данных типа *float*, соответствующих стандарту IEEE 754 (IEC 60559), смежные числа имеют смежные целочисленные представления. Такое свойство выполняется только для чисел одного знака. Это означает, что существует изоморфизм между линейными порядками целых чисел и чисел типа *float* одного знака, бинарно соответствующих друг другу. Отметим, что большие целочисленные значения (близкие к 2^{31}) попадают в область NaN (*not a number*), поэтому некоторая область значений *float* оказывается занятой различными флагами выражения *signal NaN*, однако данное обстоятельство для предложенного нами подхода не является критичным.

Изоморфизм между линейными порядками целых чисел и чисел типа *float* одного знака, бинарно соответствующих друг другу, дает возможность упаковать внутри одного значения *float* 1 бит для знака (фактически не используемый), 2 бита кодирования области и 29 бит для представления значений функции *penalty*, полученных через периметр, объем, увеличение периметра или объема.

При использовании данного подхода вычисление значений функции *penalty* реализуется выполнением следующей последовательности действий.

1. Выбор области.
2. Вычисление значения *penalty* соответствующей функции внутри области.
3. Преобразование значения в целочисленное представление.
4. Сдвиг вправо целочисленного представления на 2 бита.
5. Добавление кода области к верхним битам, сразу после знакового бита.
6. Преобразование целочисленного представления в возвращаемое значение *float*.

Предложенная функция различает многие случаи, в которых предыдущая реализация просто возвращала ноль.

Изложенный выше подход был реализован в виде патча для *PostgreSQL*. При его разработке было принято во внимание, что стандарты языков про-

граммирования (ANSI) C и C-99 не гарантируют поддержки типа представления целых чисел *float* стандарта IEEE 754, несмотря на то, что сегодня тип *float* является общепринятым. В связи с этим разработанный патч добавляет в *PostgreSQL* проверку (через *#define*) для макроса препроцессора *__STDC_IEC_559__*, которая указывает на его совместимость со стандартом IEEE 754. Если архитектура вычислителя несовместима со стандартом IEEE 754, то разработанный патч возвращается к стандартному режиму функционирования (т. е. происходит возврат значения из области 3 без битов кода области).

Функция *pack_float()*, реализующая действия 3–6 при вычислении значений функции *penalty*, обсуждена в следующем разделе.

Особенности реализации функции pack_float()

При разработке функции *pack_float()*, размещающей области и значения *penalty* в контейнере 32-битного типа *float*, были учтены следующие требования.

1. К качеству кода для проекта *PostgreSQL* — код должен соответствовать ISO C99 и проходить компиляцию без предупреждений (компиляция с флагом — *Wall*).

2. К работоспособности кода — код должен выполняться для разных архитектур процессора *PostgreSQL*, так как достаточно часто *PostgreSQL* используется в системах на базе *ARM* или *MIPS* (компиляция с флагом — *O2*).

Код первой версии программной реализации функции *pack_float* и результаты его выполнения представлены на рис. 1.

На рис. 1 видно, что разработанная программная реализация функции *pack_float* удовлетворяет требованию 2. Однако данный код не соответствует требованиям, предъявляемым к качеству кода *PostgreSQL*. Кроме того, априори понятно, что частое использование указателей может ограничить выбор возможных оптимизаций в компиляторах (за исключением компилятора *GCC*). В связи с этим во второй вариант программной реализации функции *pack_float* был введен новый тип *union*, который обеспечил более точное преобразование значений *penalty* в 32-битовый тип *float* (рис. 2).

На рис. 2 видно, что данная версия программной реализации функции *pack_float* компилируется без каких-либо предупреждений и приводит к той же сборке, как и предыдущая версия.

Программный код, представленный на рис. 2, также может быть оптимизирован. Действительно, компилятор не имеет информации о знаке *a.i*, поэтому компилятор для выражения "*a.i/4*" должен генерировать полный машинный код "деление на 4" вместо одного сдвига вправо с помощью команды 2. Однако априори известно, что "*a.i*" всегда больше или равно нулю, поэтому вместо деления можно использовать битовый сдвиг, который обеспечит более быструю сборку кода. Описанная оптимизация была включена в третью версию программной реализации функции *pack_float* (рис. 3).

```
float pack_float_v1(float actualValue, int realm) {
int realmAdjustment = *((int*)&actualValue)/4;
int realCode = realm * (INT32_MAX/4) + realmAdjustment;
return *((float*)&realCode);
}
```

Архитектура X86_64	Архитектура ARMv7
pack_float_v3: movd %xmm0, %edx leal 3(%rdx), %eax testl %edx, %edx cmovns %edx, %eax sarl \$2, %eax movl %eax, %edx movl %edi, %eax sall \$29, %eax subl %edi, %eax addl %edx, %eax movl %eax, -4(%rsp) movss -4(%rsp), %xmm0 ret	pack_float_v3: vmov r3, s0 @ int rsb r0, r0, r0, lsl #29 add r2, r3, #3 cmp r3, #0 movlt r3, r2 add r3, r0, r3, asr #2 vmov s0, r3 bx lr

Рис. 1. Реализация *pack_float* с прямой манипуляцией битовым представлением

```
typedef union { float fp; int i; } U;

float pack_float_v2(const float v, const int r) {
    const U a = { .fp = v };
    const U b = { .i = a.i/4 + r * (INT32_MAX / 4) };

    return b.fp;
}
```

Рис. 2. Реализация *pack_float* с использованием типа *union*

```
float pack_float(const float v, const int r) {

    const U a = { .fp = v };
    const U b = { .i = (a.i >> 2) + r * (INT32_MAX / 4) };

    return b.fp;
}
```

X86_64	ARMv7
pack_float: movd %xmm0, %eax movl %edi, %edx sall \$29, %edx sarl \$2, %eax subl %edi, %edx addl %edx, %eax movl %eax, -4(%rsp) movss -4(%rsp), %xmm0 ret	pack_float: vmov r3, s0 @ int rsb r0, r0, r0, lsl #29 add r0, r0, r3, asr #2 vmov s0, r0 bx lr

Рис. 3. Итоговая реализация *pack_float*

Результаты тестирования
модифицированной функции *penalty*

Были проведены тестовые испытания модифицированной функции *penalty* на виртуализированной через *Microsoft Hyper-V* системе *Ubuntu 14 LTS* с процессорами *Intel Core i5*, *SSD*-дисками и неограниченной динамической ОЗУ. Расход ОЗУ во время тестов никогда не превышал 9 ГБайт. Все тестируемые операции были последовательными и не зависели от числа процессоров.

Листинг тестового скрипта представлен на рис. 4.

```
CREATE TABLE dataTable AS SELECT cube (array[random(),random(),random()]) c FROM
generate_series(1,1e6,1);
CREATE INDEX idx ON dataTable USING gist(c);
CREATE TABLE queries(id int,l1 float,l2 float,l3 float, u1 float,u2 float, u3 float, q cube);
INSERT queries(id,l1,l2,l3) select s,random(),random(),random() FROM generate_series(1,1e4,1) s;
UPDATE queries SET q = cube(array[l1,l2,l3],array[l1+0.1,l2+0.1,l3+0.1]);

SELECT id,(SELECT count(*) FROM dataTable dt WHERE dt.c<@q) FROM queries ;

SELECT pg_size_pretty(pg_relation_size('idx'));
```

Рис. 4. Скрипт тестирования

В ходе выполнения представленного на рис. 4 запроса была реализована следующая последовательность действий.

Шаг 1. Создание таблицы данных, в которой хранится миллион *3D*-точек.

Шаг 2. Создание для этих данных индекса *GiST* с *R*-деревом.

Шаг 3. Создание 10^4 областей, используемых для геометрических поисков.

Шаг 4. Подсчет для каждой области числа точек данных, находящихся в пределах области.

Шаг 5. Вывод размера сгенерированного индекса.

При этом измеряли: время создания индекса; время вычисления поисков по запросам; размер результирующего индекса. Для сравнения производительности запросов (время выполнения в секундах) исправленного и стандартного алгоритмов было проведено десять независимых испытаний. По их результатам были вычислены оценки среднего времени выполнения теста M и среднее квадратичное отклонение времени выполнения теста SD . Проверка статистической значимости отличий данных оценок друг от друга проверяли с помощью t -теста.

Результаты проведенных тестов для запросов *SELECT* и *INSERT*, представленные на рис. 5, свидетельствуют о значительной разнице между выбранными количественными характеристиками скорости работы программных реализаций стандартной и модифицированной функций *penalty*.

На рис. 5 видно, что оценки значений выбранных показателей составили:

- для запроса *SELECT*
 - стандартная функция *penalty* — $M = 95,07$ с, $SD = 2,54$;
 - модифицированная функция *penalty* — $M = 59,74$ с, $SD = 3,52$;
- для запросов *INSERT*
 - стандартная функция *penalty* — $M = 68,88$ с, $SD = 2,09$;
 - модифицированная функция *penalty* — $M = 75,64$ с, $SD = 1,78$.

Значения t -критерия подтвердили статистическую значимость обнаруженных различий оценок времени выполнения запросов *SELECT* и *INSERT*, $p = 0,0081$ и $p = 0,0001$ соответственно.

Таким образом, время выполнения запроса *SELECT* в среднем уменьшилось на 40 %, что объясняется улучшениями, внесенными в дерево индексов вследствие усовершенствований функции *penalty*. Создание индекса модифицированным алгоритмом заняло в сравнении с используемым алгоритмом на 10 % времени больше, что объясняется выполнением центральным процессором большего числа тактов при вычислении улучшенной функции *penalty*. При этом размер индекса уменьшился с 85 до 82 Мбайт.

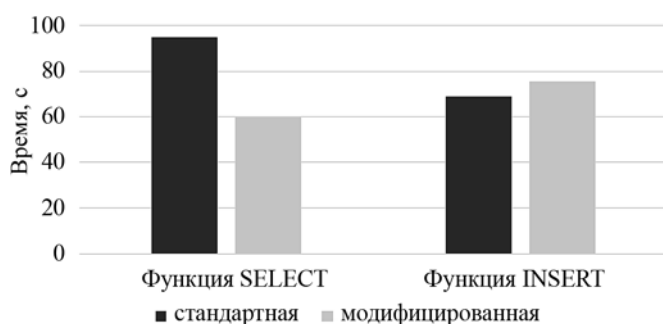


Рис. 5. Производительность поиска и вставки данных до и после применения модифицированной функции *penalty*

Заключение

Разработана модифицированная функция *penalty*, обеспечивающая повышение на 40 % производительности геометрического поиска, который выполняется с использованием технологии R -дерева в *GiST*, а также ее программная реализация в виде патча к исходному коду *PostgreSQL*.

Предложенная авторами модификация обсуждалась с сообществом *PostgreSQL* [11] и с Н. Бекманом, автором *Revised R**-дерева [10]. Итогом данных обсуждений стал консолидированный вывод о том, что, несмотря на достигнутые улучшения в производительности, требуется модификация самого *GiST API*, который не позволяет провести полноценное внедрение современных алгоритмов вставки через расчет функции *penalty* в R -дерево. В частности, современный *API GiST* не позволяет реализовать одну из наиболее важных возможностей, имеющихся в *Revised R**-дерево — оптимизацию перекрытия образующихся при вставке *bounding box*. По словам Н. Бекмана, "Оптимизация перекрытия — один из основных способов, если не основной способ настройки производительности запроса в *Revised R**-дерево. Вы возвратитесь во времена R -дерева, если это свойство будет отключено". При этом оптимизация перекрытия требует расширения *API GiST*. Таким образом, поиск решений, обеспечивающих реализацию алгоритмов оптимизации перекрытия в существующем *GiST API*, является актуальным направлением дальнейших исследований.

Список литературы

1. Bayer R. Binary b-trees for virtual memory // Proceedings of the 1971 ACM SIGFIDET Workshop on Data Description, Access and Control. ACM. 1971. P. 219—235.
2. Garcia-Molina H., Ullman J. D., Widom J. Database system implementation, vol. 654. Prentice Hall Upper Saddle River, NJ, 2000. 643 p.
3. Hellerstein J. M., Naughton J. F., Pfeffer A. Generalized search trees for database systems // VLDB'95 Proceedings of the 21th International Conference on Very Large Data Bases. 1995. P. 562—573.
4. Kornacker M., Mohan C., Hellerstein J. M. Concurrency and recovery in generalized search trees // ACM SIGMOD Record. 1997. Vol. 26. P. 62—72.
5. Chilingarian I., Bartunov O., Richter J., Sigaev T. Postgresql: The suitable dbms solution for astronomy and astrophysics // Astronomical Data Analysis Software and Systems (ADASS) XIII. 2004. Vol. 314. P. 225.
6. Guttman A. R-trees: a dynamic index structure for spatial searching // ACM. 1984. Vol. 14. P. 47—57.
7. Borodin A., Mirvoda S., Porshnev S. Database index debug techniques: A case study // In International Conference: Beyond Databases, Architectures and Structures. Springer, 2015. P. 648—658.
8. Borodin A., Kiselev Yu., Mirvoda S., Porshnev S. On design of domain-specific query language for the metallurgical industry // International Conference: Beyond Databases, Architectures and Structures. Springer, 2015. P. 505—515.
9. Borodin A. [proposal] improvement of gist page layout. URL: <https://clck.ru/C6GFi>
10. Beckmann N., Seeger B. A revised r^* -tree in comparison with related index structures // In SIGMOD'09 Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, 2009. P. 799—811.
11. Borodin A. GiST penalty functions [proof of concept]. URL: <https://clck.ru/C6GG7>

Improving Penalty Function of R-Tree Over Generalized Index Search Tree as Possible Way to Advance Performance of PostgreSQL Cube Extension

S. V. Porshnev, s.v.porshnev@urfu.ru, **O. A. Ponomareva**, o.a.ponomareva@urfu.ru, Ural Federal University, 620002, Yekaterinburg, Russian Federation, **A. M. Borodin**, amborodin@urfu.ru, "Yandex", 620014, Yekaterinburg, Russian Federation, **S. G. Mirvoda**, s.g.mirvoda@urfu.ru, "Octonika", 620014, Yekaterinburg, Russian Federation

Corresponding author:

Ponomareva Olga A., Lector, Ural Federal University, 620002, Yekaterinburg, Russian Federation
E-mail: o.a.ponomareva@urfu.ru

Received on October 16, 2017

Accepted on October 25, 2017

At present time databases solve many different technical challenges. They store data compactly, resolve data write conflicts and allow efficient data querying. The last challenge is undertaken by so called access methods (AM), i.e. algorithms and data structures enabling quick retrieval of the data by certain conditions. The paramount of AMs for modern relational databases is a B-tree allowing fast retrieval of data by the primary key and many other useful operations. Generalized index search tree (GiST) is an AM technique, which allows abstraction of significant parts of the data access methods, structured as a balanced tree. Use of the GiST allows AM developer to concentrate on their own case-specific details of AM and skip common work on the tree structure implementation within database engine, a query language integration, a query planner support, etc. Generalized index search tree (GiST) greatly simplifies data access methods development. Important parts like query processing, failure recovery, memory management are implemented in generic code, so access method developer has to implement only specifics of the desired algorithm. But this generality comes with a significant cost of fitting a method for GiST API. In this paper we present a number of tweaks to the R-tree implementation inside PostgreSQL GiST framework and analyze possible ways of GiST API advancements.

Keywords: generalized index search tree, multidimensional access method, PostgreSQL, R-tree

For citation:

Porshnev S. V., Ponomareva O. A., Borodin A. M., Mirvoda S. G. Improving Penalty Function of R-tree over Generalized Index Search Tree as Possible Way to Advance Performance of PostgreSQL Cube Extension, *Programmnaya Ingeneria*, 2018, vol. 9, no 1, pp. 22–28.

DOI: 10.17587/prin.9.22-28

References

1. **Bayer R.** Binary b-trees for virtual memory, *Proceedings of the 1971 ACM SIGFIDET Workshop on Data Description, Access and Control*, ACM, 1971, pp. 219–235.
2. **Garcia-Molina H., Ullman J. D., Widom J.** *Database system implementation*, vol. 654, Prentice Hall Upper Saddle River, NJ, 2000, 643 p.
3. **Hellerstein J. M., Naughton J. F., Pfeffer A.** Generalized search trees for database systems, *VLDB'95 Proceedings of the 21th International Conference on Very Large Data Bases*, 1995, pp. 562–573.
4. **Kornacker M., Mohan C., Hellerstein J. M.** Concurrency and recovery in generalized search trees, *In ACM SIGMOD Record*, ACM, 1997, vol. 26, pp. 62–72.
5. **Chilingarian I., Bartunov O., Richter J., Sigaev T.** PostgreSQL: The suitable dbms solution for astronomy and astrophysics, *In Astronomical Data Analysis Software and Systems (ADASS) XIII*, 2004, vol. 314, pp. 225.
6. **Guttman A.** R-trees: a dynamic index structure for spatial searching, *ACM*, 1984, vol. 14, pp. 47–57.
7. **Borodin A., Mirvoda S., Porshnev S.** Database index debug techniques: A case study, *In International Conference: Beyond Databases, Architectures and Structures*, Springer, 2015, pp. 648–658.
8. **Borodin A., Kiselev Yu., Mirvoda S., Porshnev S.** On design of domain-specific query language for the metallurgical industry, *International Conference: Beyond Databases, Architectures and Structures*, Springer, 2015, pp. 505–515.
9. **Borodin A.** [proposal] improvement of gist page layout, available at: <https://clck.ru/C6GFf>
10. **Beckmann N., Seeger B.** A revised r*-tree in comparison with related index structures, *In SIGMOD '09 Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 799–811.
11. **Borodin A.** GiST penalty functions [proof of concept], available at: <https://clck.ru/C6GG7>

С. Ю. Болотова, канд. физ.-мат. наук, доц., e-mail: bolotova.svetlana@gmail.com,

А. В. Зонов, преподаватель, e-mail: andryzonov@gmail.com,

А. П. Тютин, преподаватель, e-mail: yutin.ap@gmail.com,

Воронежский государственный университет

Навигация внутри помещений в мобильных приложениях

Навигация внутри помещений является достаточно сложной задачей, особенно в условиях больших многоэтажных зданий с множеством стен. Кроме этого, требуются высокая точность и скорость определения местоположения. В работе описаны основные идеи реализации мобильного приложения, позволяющего строить оптимальные маршруты внутри помещений с помощью Bluetooth-маячков и удовлетворяющего отмеченным выше требованиям.

Ключевые слова: навигация, bluetooth-маячки, beacons, мобильные приложения, местоположение, трилатерация, триангуляция, тесселяция, построение маршрутов

Введение

На настоящее время одной из актуальных тем в мобильных разработках является навигация внутри помещений. Это обстоятельство связано с тем, что здания становятся все более объемными и нередко имеют довольно сложную структуру, что затрудняет ориентацию в пространстве. Кроме того, решения, применяемые в навигации внутри помещений, помогают и в ориентировании вне зданий, на улице — там, где в условиях плотной застройки использование систем спутниковой навигации затруднено (нет спутников в прямой видимости, присутствует только отраженный/ослабленный/зашумленный сигнал GPS/Глонасс и т. д.).

Навигация внутри помещений имеет следующий ряд особенностей.

1. Местоположение внутри помещений необходимо определять быстро и с высокой точностью, чего не позволяет навигация GPS/ГЛОНАСС.

2. Присутствует такой фактор, как этажность здания.

3. В помещении, как правило, нет дорог, по которым можно было бы прокладывать маршруты.

Таким образом, узловыми моментами для решения задачи навигации внутри помещений являются вопросы определения местоположения с высокой точностью и построения маршрутов на открытой местности. Целью настоящей работы стали подбор и реализация алгоритмов программных систем, позволяющих найти ответы на поставленные задачи.

В ходе изучения рассматриваемой предметной области был сделан вывод, что поставленная задача на настоящий момент является актуальной, а работающих решений, удовлетворяющих необходимым требованиям, не существует.

Определению местоположения с помощью технологии Bluetooth (Bluetooth Indoor Positioning) посвя-

щена работа [1] Anja Bakkeliën из Женевского университета. Однако ее задача заключалась в разработке подмодуля для глобального модуля позиционирования (GPM), в основе которого работает совокупность сенсоров. В настоящей работе ставится условие реализации самодостаточной навигационной системы, использующей в своей основе технологию iBeacons.

Технология iBeacons, представленная Apple в iOS 7, расширила возможности библиотеки для работы со службами геолокации. Вместо геолокации по широте и долготе iBeacon использует низкоэнергетический сигнал Bluetooth, который обнаруживается телефоном. Соответствующий формат данных строго типизирован и детально описан на официальном сайте Apple для разработчиков.

Построение маршрутов на открытой местности изучала Anita Graser из Австрийского технологического института. В ее работе [2] рассмотрены вопросы внедрения работы с открытыми пространствами в OpenStreetMap — веб-картографическом проекте по созданию географической карты мира. Однако описанные в указанной работе алгоритмы не учитывают препятствий на открытых пространствах, а реализация не учитывает специфики мобильных платформ.

Технология Bluetooth Beacons

Для определения местоположения мобильного устройства нередко используются Bluetooth-маячки (beacons).

Bluetooth-маячки — это обычные Bluetooth 4.0 LE (Low Energy) устройства, т. е. их роль может с успехом выполнять любое устройство, оснащенное чипом BLE (Bluetooth Low Energy), например, смартфон iPhone. Для этой цели необходимо установить маячки по периметру помещения, а координаты их расположения сохранить в базе данных. Маячки с заданной

периодичностью будут осуществлять широкополосную рассылку, содержащую идентифицирующую их информацию. Пользовательское приложение, циклично получая эти сигналы, сможет с помощью базы данных определить координаты маячков. Далее на основе анализа интенсивности посылаемого сигнала RSSI (*Received Signal Strength Indicator*) может быть вычислено местоположение пользователя [3].

Алгоритм определения местоположения

Получаемый параметр RSSI позволяет определить удаленность маячка от приемника сигнала — смартфона. Это осуществляется делением текущего значения RSSI на эталонное значение интенсивности, измеренное на расстоянии 1 м. Очевидно, что значение RSSI обратно пропорционально удаленности смартфона от маячка. Однако вследствие физического эффекта интерференции сигналов значение RSSI между двумя стационарными объектами непостоянно (рис. 1, 2).

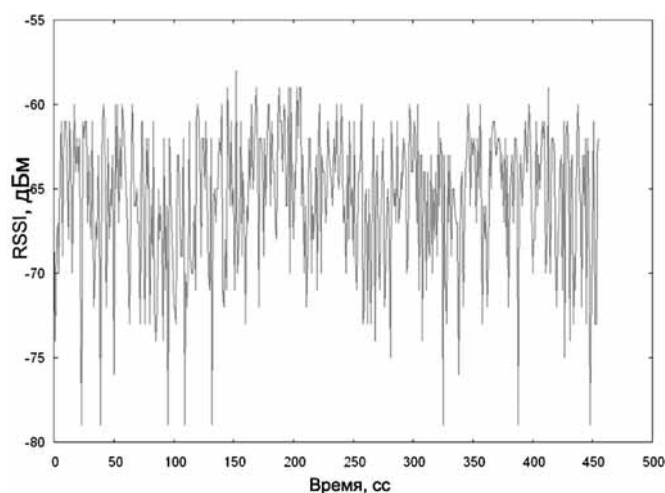


Рис. 1. Значения RSSI между маячком и устройством на фиксированном расстоянии 3 м

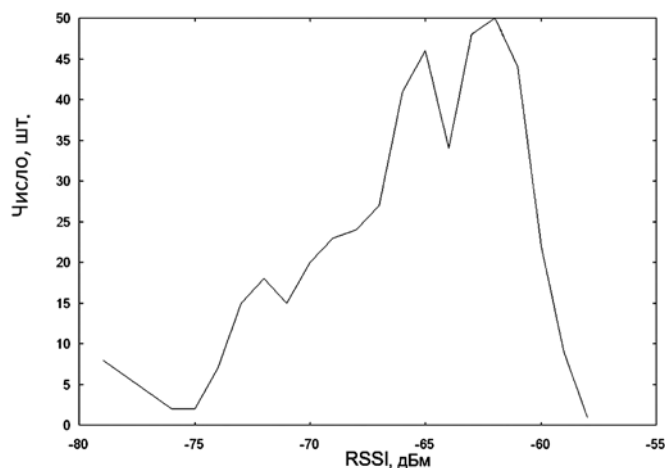


Рис. 2. Гистограмма значений для графика, показанного на рис. 1

Как видно представленных графиках, при использовании единственного маячка можно получать разную удаленность. Графически ситуацию можно представить окружностью с центром в точке с координатами расположения маячка и радиусом, равным вычисленному расстоянию до него. Радиус этой окружности даже при неподвижном состоянии приемника будет меняться с течением времени. Используя информацию от двух маячков, можно получить две окружности с изменяющимися радиусами и, возможно, одной или двумя точками пересечения. Если маячки дают точную информацию об удаленности, то используя два маячка, точное местоположение можно рассчитать лишь находясь на одной линии с ними. При отклонении от этой линии получатся две точки пересечения и две возможные координаты. Однако, поскольку значение RSSI колеблется, точек пересечения может и не быть. Отсюда можно сделать вывод, что для однозначного определения местоположения необходимо как минимум три маячка.

Алгоритм нахождения координат по трем или более точкам и удалению от них называется алгоритмом трилатерации [4]. Его графическое представление дано на рис. 3.

Задача трехмерной трилатерации решается путем нахождения координат пересечения трех сфер, которые определяют решением системы уравнений. Вначале составим систему уравнений для трех сфер. Поскольку любые три точки, не лежащие на одной прямой, определяют в пространстве плоскость, поместим начало координат этой плоскости в центр

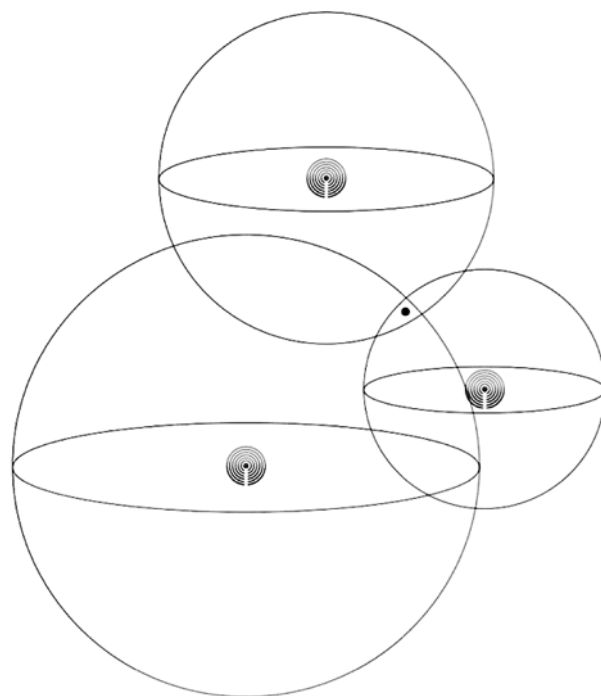


Рис. 3. Алгоритм трилатерации — графическое представление

одной из сфер, а центр второй сферы поместим на ось координат Ox :

$$\begin{aligned}r_1^2 &= x^2 + y^2 + z^2, \\r_2^2 &= (x - d)^2 + y^2 + z^2, \\r_3^2 &= (x - i)^2 + (y - j)^2 + z^2,\end{aligned}$$

где r_1, r_2, r_3 — радиусы сфер; d — расстояние между центрами первой и второй сфер по оси Ox ; i — расстояние между центрами первой и третьей сфер по оси Ox ; j — расстояние между центрами первой и третьей сфер по оси Oy .

В точке (x, y, z) будет располагаться приемник сигналов:

$$\begin{aligned}x &= \frac{r_1^2 - r_2^2 + d^2}{2d}, \\y &= \frac{r_1^2 - r_3^2 - x^2 + (x - i)^2 + j^2}{2j} = \\&= \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x, \\z &= \pm \sqrt{r_1^2 - x^2 - y^2}.\end{aligned}$$

Таким образом, получены уравнения для вычисления координат (x, y, z) приемника. Очевидно, координата высоты z может принимать два значения. Это обстоятельство обусловлено тем, что передатчики и приемник могут находиться не в одной плоскости. Для нашей задачи можно рассматривать только нижнее значение, так как маячки располагаются выше пользовательских устройств.

Благодаря тому требованию, что в рассматриваемой системе все маячки должны лежать в одной плоскости, для решения поставленной задачи положением на вертикальной оси можно пренебречь. Устройство пользователя — приемник находится в другой плоскости. Однако в силу плотного покрытия области маячками и фильтрации удаленных маячков можно считать, что приемник находится в той же плоскости.

В нашем случае координаты вычисляют в помещениях со стенами, столбами и электронными устройствами. Такие помещения характеризуются наличием "шума". Представим области действия маячков в виде окружностей, где центр окружности — точка расположения маячка, а радиус определяется силой сигнала. Пересечением подобных окружностей может являться область, пустая или нет. Поэтому целесообразно модифицировать рассматриваемый алгоритм, воспользовавшись методом итеративной трилатерации [5]. Суть этого алгоритма состоит в итеративной минимизации ошибки:

$$|f_i| = \left| \frac{d_i - \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2}}{d_i} \right|,$$

где $|f_i|$ — вещественное значение, характеризующее ошибку, а (x_e, y_e) — тривиальная начальная оценка. Необходимо итеративно высчитывать значение $|f_i|$ до тех пор, пока ошибка не уменьшится до заданного значения.

Однако эмпирические наблюдения показывают, что в результате все еще остается "шум", который проявляется как изменение координат пользователей при отсутствии физического движения. Фильтрация такого "шума" может быть выполнена с помощью вмонтированных в устройства сенсоров. Для решения этой части задачи подходят методы на основе рекурсивной оценки Байеса.

С указанной целью в рамках настоящей работы были рассмотрены возможности применения фильтра Калмана [6] и многочастотного фильтра [7]. Фильтр Калмана предназначен для рекурсивного дооценивания вектора состояния априорно известной динамической системы. Для расчета текущего состояния системы необходимо знать текущее измерение, а также предыдущее состояние самого фильтра. Алгоритм работает в два этапа. На этапе прогнозирования фильтр Калмана экстраполирует значения переменных состояния, а также их неопределенности. На втором этапе по данным измерения (полученного с некоторой погрешностью) результат экстраполяции уточняется. Благодаря пошаговой природе алгоритма он может в реальном времени отслеживать состояние объекта [8].

Многочастотный фильтр основан на последовательном методе Монте-Карло. Это рекурсивный алгоритм численного решения задач оценивания (фильтрации, сглаживания), особенно актуальный для нелинейных и негауссовских случаев. В отличие от фильтра Калмана, многочастотные фильтры не зависят от методов линеаризации или аппроксимации.

Экспериментальным путем было установлено, что из двух рассмотренных вариантов фильтр Калмана обеспечивает лучшее сглаживание.

Построение маршрутов на открытой местности

Перейдем к вопросу построения маршрутов. Поскольку здание обычно представляется не набором коридоров/лабиринтов, а некоторым числом помещений, то, как было сказано ранее, изначально отсутствует заданное множество дорог. Процесс решения рассматриваемой задачи можно разбить на следующие два этапа.

1. Представив помещение в виде полигона, провести его триангуляцию.

2. Приняв вершины получившихся треугольников за вершины графа, а их стороны — за его ребра, определить ближайшие к стартовой и конечной точкам вершины и найти кратчайший путь от одной до другой.

По определению, простой полигон — упорядоченная последовательность точек $V_0, \dots, V_n$. Соседние вер-

шины соединены ребром $\langle V_i, V_{i+1} \rangle$, где $0 \leq i \leq n - 2$.

Ребро $\langle V_{n-1}, V_0 \rangle$ соединяет первую и последнюю вершины. Каждая вершина принадлежит ровно двум ребрам. Единственное место пересечения любых двух ребер — вершина полигона.

Если полигон простой, то во время прохода по его ребрам внутренняя область всегда находится с одной и той же стороны. Полигон будем считать направленным против часовой стрелки, если при проходе вдоль ребер внутренняя область находится слева.

Триангуляция — это разбиение геометрического объекта на симплексы, т. е. n -мерные обобщения треугольника. В нашем случае объект находится на плоскости, соответственно, полигон разбивается на треугольники.

Для триангуляции существует множество алгоритмов. Рассмотрим один из них (полностью описан в работе [9]), называемый *ear clipping* (отсекание "ушей", "ушная" триангуляция). Сложность данного алгоритма — $O(n^2)$.

"Ухом" полигона называется треугольник, образованный тремя последовательными вершинами $V_{i_0}, V_{i_1}, V_{i_2}$, где V_{i_1} — выпуклая вершина (ее внутренний угол меньше 180°), отрезок $\langle V_{i_0}, V_{i_2} \rangle$ полностью лежит внутри полигона, при этом ни одна вершина полигона, кроме $V_{i_0}, V_{i_1}, V_{i_2}$, не принадлежит треугольнику. Отрезок $\langle V_{i_0}, V_{i_2} \rangle$ называется диагональю полигона. Вершина V_{i_1} называется "кончиком уха". Треугольник (полигон, для которого $n = 32$) состоит из одного "уха", однако "кончиком уха" можно считать любую из трех вершин. Полигон с четырьмя и более сторонами всегда имеет как минимум два непересекающихся "уха".

Предлагается рекурсивный подход к решению задачи "ушной" триангуляции. Если возможно найти "уха" в полигоне с $n \geq 4$ вершинами и "отрезать" его, то останется полигон из $n - 1$ вершин и можно повторить процесс. Такой алгоритм характеризуется сложностью $O(n^3)$.

Однако существует модификация этого алгоритма, приводящая к сложности $O(n^2)$ [9]. Первый шаг заключается в том, чтобы хранить полигон в двусвязном кольцевом списке, что дает возможность быстрого удаления кончиков ушей. Создание такого списка реализуется алгоритмом со сложностью $O(n)$. Вторым шагом является проход по всем вершинам и поиск "ушей". Для каждой выпуклой вершины V_i возьмем треугольник $\langle V_{i-1}, V_i, V_{i+1} \rangle$ и проверим, существует ли среди остальных вершин полигона такая, которая лежит в этом треугольнике. Если ни одна из них не лежит внутри треугольника, треугольник будет являться "ухом".

Еще одним шагом оптимизации рассматриваемого алгоритма может быть использование следующего факта. Очевидно, что если в проверяемый треугольник попадает какая-либо другая вершина многоугольника, то существует невыпуклая вершина, находящаяся внутри этого треугольника. Таким

образом, проверять на нахождение внутри треугольника нужно только те вершины, углы которых являются невыпуклыми (внутренний по отношению к полигону угол больше 180°).

Структура данных для полигона состоит из четырех двусвязных списков:

- вершины полигона хранятся в циклическом списке;
- выпуклые вершины хранятся в линейном списке;
- невыпуклые вершины хранятся в линейном списке;
- "кончики ушей" хранятся в циклическом списке.

Создав первые три списка, можно начать удалять "уши" по одному. Если V_i — удаляемая из полигона вершина, описание углов вершин V_{i-1} и V_{i+1} может измениться следующим образом [9]:

- если один из этих углов выпуклый, то он остается выпуклым;
- если один из углов V_j принадлежал другому "уху", то после удаления "уха" V_i "уха" V_j может исчезнуть;
- если один из углов невыпуклый, он может стать выпуклым, а также может создать новое "уха".

Таким образом, после удаления "уха" V_i , если соседняя вершина выпуклая, необходимо проверить, не образует ли она новое "уха", обойдя невыпуклые вершины и проверив принадлежность этих вершин треугольнику "уха". Всего есть $O(n)$ "ушей". Каждое обновление соседних вершин влечет проверку, занимающую $O(n)$ операций. Таким образом, весь процесс "отрезания ушей" представляет собой алгоритм со сложностью $O(n^2)$.

"Ушной" алгоритм также может быть применен к полигонам с дырками. Дыркой называется полигон, полностью лежащий внутри триангулируемого полигона. В таком случае перед применением алгоритма триангуляции необходимо соединить внешний полигон и все его дырки в одну последовательность точек.

Представим себе полигон с одной дыркой. Он состоит из внешнего и внутреннего полигонов. Порядок вершин во внешнем и внутреннем полигонах должен быть противоположным. Задача заключается в том, чтобы найти во внешнем и внутреннем полигонах вершины, между которыми можно построить отрезок, не пересекающий ни с одним из других отрезков полигонов. В работе [9] описан один из способов нахождения таких вершин, суть которого представляется следующим алгоритмом.

Шаг 1. Найти точку M внутреннего полигона с максимальным значением по оси x .

Шаг 2. Найти самую ближнюю к точке M точку пересечения I луча $M + t(1, 0)$ со сторонами внешнего полигона.

Шаг 3. Если I — вершина внешнего полигона, искомая точка найдена.

Шаг 4. Иначе I — точка отрезка $\langle V_i, V_{i+1} \rangle$. Выбрать точку P — конец отрезка $\langle V_i, V_{i+1} \rangle$ с большим значением по оси x .

Шаг 5. Проверить невыпуклые вершины внешнего полигона (кроме P) на содержание внутри треугольника $\langle M, I, P \rangle$. Если ни одна из них не лежит внутри треугольника, то точка P является искомой.

Шаг 6. Иначе хотя бы одна невыпуклая вершина лежит внутри треугольника. Среди этих вершин выбрать такую вершину R , что угол IMR минимален. Если таких точек несколько, то все они лежат на одном луче, выходящем из M , и необходимо выбрать ближайшую к M точку. Точка R является искомой.

Вершины M и P необходимо дублировать и соединить полигоны в один двумя отрезками. Полученный полигон можно триангулировать как обычный полигон без дырок.

Полигон может иметь несколько дырок. В таком случае нужно применить описанный выше алгоритм к каждой из этих дырок последовательно, выбирая из списка оставшихся такую, которая содержит вершину с максимальным значением x .

Было замечено, что при малом числе углов полигона после триангуляции может получиться достаточно разреженный граф, что не позволит построить оптимальный маршрут через его вершины. Поэтому было решено провести уплотнение полигональной сетки. Это можно сделать с помощью *тесселяции*. Тесселяция — это процесс разбиения полигональной сетки на более мелкие многоугольники. В результате тесселяции граф получается более плотным, и кратчайшие пути становятся гораздо более естественными для движения пешехода.

Итак, для выполнения триангуляции данные, описывающие карту помещения, должны представляться набором полигонов, где координаты углов внешних стен помещения перечислены по часовой стрелке, а координаты углов внутренних стен и других преград (дырок в полигоне) — против часовой стрелки, или наоборот.

Поскольку помещение может представлять невыпуклый многоугольник, а внутри него могут оказаться дополнительные преграды, необходим алгоритм триангуляции, учитывающий эти два обстоятельства.

На выходе алгоритма триангуляции и, если нужно, тесселяции, должен получиться набор вершин треугольников. Далее этот набор необходимо преобразовать в неориентированный взвешенный граф: вершины треугольников становятся вершинами графа, а стороны — его ребрами. При этом длина стороны становится весом соответствующего ребра.

Для построения маршрута достаточно использовать один из алгоритмов поиска кратчайшего пути, например, алгоритм Дейкстры.

Применяя описанные выше операции, получим маршрут, который представляется на карте ломаной линией. В качестве пути дальнейшего улучшения результатов можно рассмотреть сглаживание полу-

ченного маршрута, например, с помощью кубических сплайнов.

Заключение

Результатом представленного исследования стал работоспособный прототип приложения, который удовлетворяет требованиям к производительности, точности определения местоположения, а также умеет строить оптимальные маршруты внутри помещений. На основе проведенных экспериментов были получены следующие результаты.

- Описанные алгоритмы обладают достаточной производительностью, чтобы проводить все расчеты в реальном времени с учетом ограничения мобильных устройств, на которых прототип тестировался.

- Погрешность вычислений координат составляет порядка 15 м. Для ее уменьшения необходимо учитывать, в первую очередь, внутренние датчики.

- Погрешность увеличивается при движении и поворотах. Поэтому в текущей реализации невозможно избежать небольших "подергиваний" точки, обозначающей местоположение пользователя на карте при движениях такого типа.

- Удастся построить достаточно правдоподобный, оптимальный маршрут в помещении, не имеющем определенных дорог.

Список литературы

1. **Bekkeliën A., Deriaz M., Marchand-Maillet S.** Bluetooth indoor positioning. Master's thesis, University of Geneva. 2012. 49 p.
2. **Graser A.** Integrating Open Spaces Into OpenStreetMap Routing Graphs for Realistic Crossing Behavior in Pedestrian Navigation // *GI_Forum — Journal for Geographic Information Science*. 2016. N. 1. P. 217—230.
3. **Wu B.-F., Jen C.-L., Chang K. C.** Neural fuzzy based indoor localization by Kalman filtering with propagation channel modeling // In *IEEE International Conference on Systems, Man and Cybernetics*. 2007. P. 812—817.
4. **Lau E.-E. L., Chung W.-Y.** Enhanced RSSI-based real-time user location tracking system for indoor and outdoor environments // In *ICCIT 07: Proceedings of the 2007 International Conference on Convergence Information Technology*. Washington, DC, USA. 2007. P. 1213—1218.
5. **Thrun S., Burgard W., Fox D.** Probabilistic Robotics. Early Draft-MIT Press. 2005. 492 p.
6. **Chen Z.** Bayesian filtering: From Kalman filters to particle filters, and beyond. Statistics. McMaster University, Hamilton, 2003. 69 p.
7. **Rekleitis I. M.** A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02. Centre for Intelligent Machines. University St., Montreal, Quebec, Canada H3A 2A. 2004. 36 p.
8. **Carpenter J., Clifford P., Fernhead P.** An improved particle filter for non-linear problems // *EE Proceedings-Radar, Sonar and Navigation*. 1999. P. 2—7.
9. **Eberly D.** Triangulation by ear clipping. Geometric Tools, LLC. 2015. 13 p.

Indoor Navigation in Mobile Applications

S. Yu. Bolotova, bolotova.svetlana@gmail.com, **A. V. Zonov**, andryzonov@gmail.com,
A. P. Tutin, tyutin.ap@gmail.com, The Voronezh State University, Voronezh,
394018, Russian Federation

Corresponding author:

Bolotova Svetlana Yu., Associate Professor, The Voronezh State University, 394018, Voronezh, Russian Federation
E-mail: bolotova.svetlana@gmail.com

Received on October 2, 2017

Accepted on October 16, 2017

The present article describes basic ideas of implementation of mobile application dedicated to indoor navigation. Indoor navigation is a complex issue, especially when it comes to large buildings with lots of rooms and rich infrastructure. The goal of the research was to achieve high accuracy and performance of mobile device positioning and indoor routes calculation.

Bluetooth beacons were suggested to determine mobile device location. Analyzing the received signal strength indicator (RSSI) and beacons position on the map provided an ability to calculate the distance from each beacon to the mobile device.

Trilateration algorithms were composed with Kalman filter which allowed for fast and precise device positioning.

To calculate pedestrian-friendly indoor routes the authors addressed game development methods: room map was suggested to be in a form of a polygon with holes, which was decomposed via triangulation and tessellation algorithms. Among all triangulation algorithms that work with "holed" polygons the modification of Earcut triangulation algorithm was considered to be one of the most suitable and efficient. As a result, getting a polygonal mesh of high density allowed to use this mesh as a graph and apply any path searching algorithm to calculate the determined route.

The result of the whole research was a working prototype of a mobile application which satisfied performance and accuracy requirements. Preferred algorithms have good potential and can be used as a basis for future improvements. Inaccuracy of measurements should always be considered and its handling is a self-sufficient problem.

Keywords: navigation, Bluetooth beacons, beacons, mobile applications, location, trilateration, triangulation, tessellation, route building

For citation:

Bolotova S. Yu., Zonov A. V., Tutin A. P. Indoor Navigation in Mobile Applications, *Programmnaya Ingeneria*, 2018, vol. 9, no. 1, pp. 29–34.

DOI: 10.17587/prin.9.29-34

References

1. **Bekkelien A., Deriaz M., Marchand-Maillet S.** *Bluetooth indoor positioning*, Master's thesis, University of Geneva, 2012, 49 p.
2. **Graser A.** Integrating Open Spaces Into OpenStreetMap Routing Graphs for Realistic Crossing Behavior in Pedestrian Navigation, *GI_Forum — Journal for Geographic Information Science*, 2016, no. 1, pp. 217–230.
3. **Wu B.-F., Jen C.-L., Chang K. C.** Neural fuzzy based indoor localization by Kalman filtering with propagation channel modeling, *In IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 812–817.
4. **Lau E.-E. L., Chung W.-Y.** Enhanced RSSI-based real-time user location tracking system for indoor and outdoor environments, *In ICCIT 07: Proceedings of the 2007 International Conference on Convergence Information Technology*, Washington, DC, USA, 2007, pp. 1213–1218.
5. **Thrun S., Burgard W., Fox D.** *Probabilistic Robotics*, Early Draft-MIT Press, 2005, 492 p.
6. **Chen Z.** *Bayesian filtering: From Kalman filters to particle filters, and beyond*, Statistics, McMaster University, Hamilton, 2003, 69 p.
7. **Rekleitis I. M.** A particle filter tutorial for mobile robot localization, Technical Report TR-CIM-04-02, Centre for Intelligent Machines, University St., Montreal, Quebec, Canada H3A 2A, 2004, 36 p.
8. **Carpenter J., Clifford P., Fernhead P.** An improved particle filter for non-linear problems, *EE Proceedings-Radar, Sonar and Navigation*, 1999, pp. 2–7.
9. **Eberly D.** *Triangulation by ear clipping*, Geometric Tools, LLC, 2015, 13 p.

С. С. Попович, канд. техн. наук, науч. сотр., e-mail: pss1@mail.ru,
Научно-исследовательский институт механики Московского государственного
университета имени М. В. Ломоносова

Особенности автоматизации эксперимента и обработки результатов при исследовании теплообмена в сверхзвуковом потоке сжимаемого газа

Представлены методика и программное обеспечение, разработанное в среде LabVIEW, для автоматизации теплофизического эксперимента на сверхзвуковой аэродинамической установке и обработки экспериментальных данных. Исследование параметров теплообмена проводили на нестационарном режиме в процессе запуска аэродинамической установки до выхода на равновесный тепловой режим. Число Маха набегающего потока составляло 2,3, критерий Рейнольдса по длине динамического пограничного слоя — не менее $2 \cdot 10^7$ на срезе сопла. Представлены графики изменения в процессе запуска установки давления и температуры торможения, температуры стенки модели, статического давления, адиабатной температуры стенки, теплового потока и числа Стантона, которые характеризуют процессы теплообмена на этапе запуска аэродинамической установки.

Ключевые слова: LabVIEW, аэродинамическая установка, виртуальный прибор, сверхзвуковой поток, адиабатная температура стенки, коэффициент восстановления температуры, коэффициент теплоотдачи

Введение

Одной из важнейших задач при разработке высокоскоростных летательных аппаратов и конструировании проточной части энергетических установок является точность переноса данных, полученных на аэродинамических трубах, на реальные условия эксплуатации. Для этого, в первую очередь, необходимо проведение тщательных экспериментальных исследований в широком диапазоне изменения параметров потока с использованием современных средств автоматизации эксперимента [1], обеспечивающих возможность многократного фиксирования данных.

Например, одной из возможных технологий экстраполяции результатов сверхзвукового турбулентного аэродинамического нагрева является прямое сравнение экспериментального и расчетного значений безразмерного коэффициента теплоотдачи — критерия подобия Стантона — на режимах, реализуемых в аэродинамической трубе [2]. Далее проводится корреляция полученных отклонений с вычислением корректирующих поправок, и в дальнейшем использование этих поправок при расчете числа Стантона уже на реально реализуемых режимах в полете или при эксплуатации наземных энергоустановок.

Особенностью обтекания тел сверхзвуковым потоком сжимаемого газа является возникновение значительного аэродинамического нагрева вследствие

большого градиента температуры в пограничном слое нежели при дозвуковых скоростях течения. Градиент температуры возникает в силу перехода скоростного напора в теплоту под воздействием вязкого трения [3]. На рис. 1 показаны возможные профили температуры в пограничном слое при адиабатической (теплоизолированной), нагретой и охлажденной стенке.

Данные по теплообмену, как правило, представляются в виде коэффициента теплоотдачи α , который в общем случае определяется отношением удельного теплового потока в стенку к температурному напору между мгновенной температурой обтекаемой стенки и некоторой определяющей температурой в потоке [4]:

$$\alpha = \frac{q_w}{T_{def} - T_w}, \quad (1)$$

где α — коэффициент теплоотдачи, Вт/(м²·К); q_w — удельный тепловой поток в стенку, Вт/м²; T_w — температура обтекаемой стенки, К; T_{def} — определяющая температура в потоке, К.

В качестве определяющей может выступать термодинамическая температура в потоке, температура торможения, начальная температура (на входе в канал с источниками или стоками теплоты) или адиабатная температура стенки. Однако как показано в ряде работ [5–8], только использование адиабатной

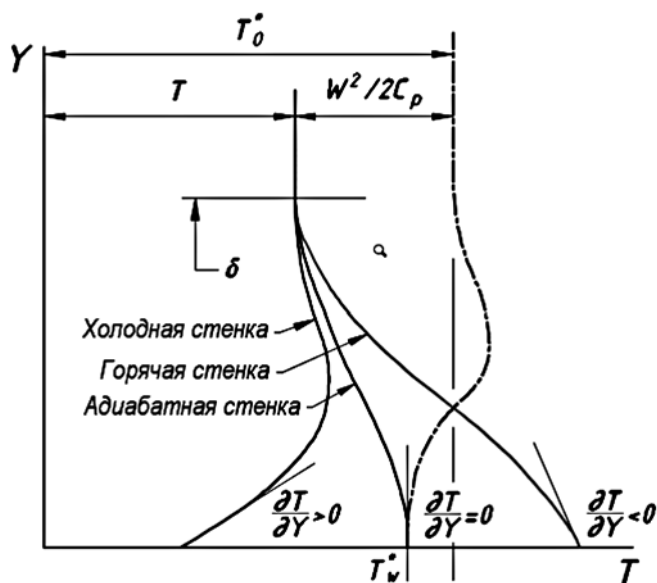


Рис. 1. Профили статической температуры T и температуры торможения T_0^* в пограничном слое сверхзвукового потока сжимаемого газа при предварительно охлажденной, нагретой или адиабатной стенке:

T — статическая (термодинамическая) температура в потоке, К; T_0^* — температура торможения потока, К; W — скорость потока, м/с; C_p — удельная теплоемкость газа при постоянном давлении, Дж/(кг·К); Y — поперечная координата, м; δ — толщина пограничного слоя, мм

температуры стенки гарантирует точность переноса данных, полученных на аэродинамических трубах, на реальные условия работы машины.

Отношение интенсивности теплоотдачи к полной потенциально возможной теплоотдаче потока газа определяется безразмерным критерием теплоотдачи — числом Стантона. Данные по теплообмену, представленные в виде числа Стантона, менее чувствительны к изменению числа Маха потока, равно как и к изменению температуры стенки, чем размерный коэффициент теплоотдачи [4]:

$$St = \frac{\alpha}{\rho_\infty w_\infty C_p} = \frac{q_w}{\rho_\infty w_\infty C_p (T_w^* - T_w)}, \quad (2)$$

где St — число Стантона; ρ_∞ — плотность газа во внешней части потока, кг/м³; w_∞ — скорость течения газа во внешней части потока, м/с.

При расчете числа Стантона по соотношению (2) одной из главных задач является определение адиабатной температуры стенки. В практике инженерных и научных расчетов эта температура определяется по следующей формуле:

$$T_w^* = T_0^* \frac{1 + r \frac{k-1}{2} M^2}{1 + \frac{k-1}{2} M^2}. \quad (3)$$

где r — коэффициент восстановления температуры; M — число Маха в потоке; k — показатель адиабаты газа.

Коэффициент восстановления температуры r показывает долю кинетической энергии потока, переходящей в теплоту на стенке. Многочисленные экспериментальные исследования, проведенные для воздуха [9], показали, что для развитого турбулентного режима течения в сверхзвуковом потоке на пластине значение r лежит в диапазоне 0,875...0,890. В литературе приведены исследования влияния на коэффициент восстановления температуры таких факторов, как число Прандтля рабочего тела [10–16], вдув/отсос пограничного слоя [17–21], форма и рельеф обтекаемой поверхности [22–28], воздействие скачков уплотнения и отрывных течений [29–37]. Результаты исследований различных авторов показывают, что коэффициент восстановления может значительно отличаться от теоретического значения для пластины.

Чувствительность коэффициента теплоотдачи к изменению коэффициента восстановления температуры дополнительно усиливается невысокими значениями температуры потока при проведении исследований на большинстве аэродинамических труб, поскольку разность температур $(T_w^* - T_w)$ в этом случае, как правило, мала. Поэтому параметры теплообмена в сверхзвуковых аэродинамических трубах могут значительно изменяться, если небольшие неточности появляются при расчете адиабатной температуры стенки.

Таким образом, точность параметров теплообмена определяется точностью, с которой удастся найти коэффициент восстановления температуры. Если рабочий диапазон аэродинамической трубы позволяет создать большой перепад температур в процессе эксперимента, то неточности определения значения величины T_w^* меньше сказываются на погрешности коэффициента теплоотдачи. Тем не менее большинство сверхзвуковых аэродинамических труб не имеют собственной системы подогрева воздуха в форкамере, что придает особое значение технологии определения коэффициента восстановления температуры и адиабатной температуры стенки.

Целью данной работы является отработка методики автоматизированного экспериментального исследования тепловых параметров сверхзвукового потока на нестационарном режиме в процессе запуска аэродинамической установки до выхода на равновесный тепловой режим. Задача обусловлена необходимостью повышения точности экстраполяции экспериментальных данных по теплообмену на реальные условия эксплуатации энергодвигательных установок.

1. Оборудование экспериментального стенда

Экспериментальные исследования проводили на сверхзвуковой аэродинамической установке непрерывного действия в научно-исследовательском Институте механики МГУ им. Ломоносова

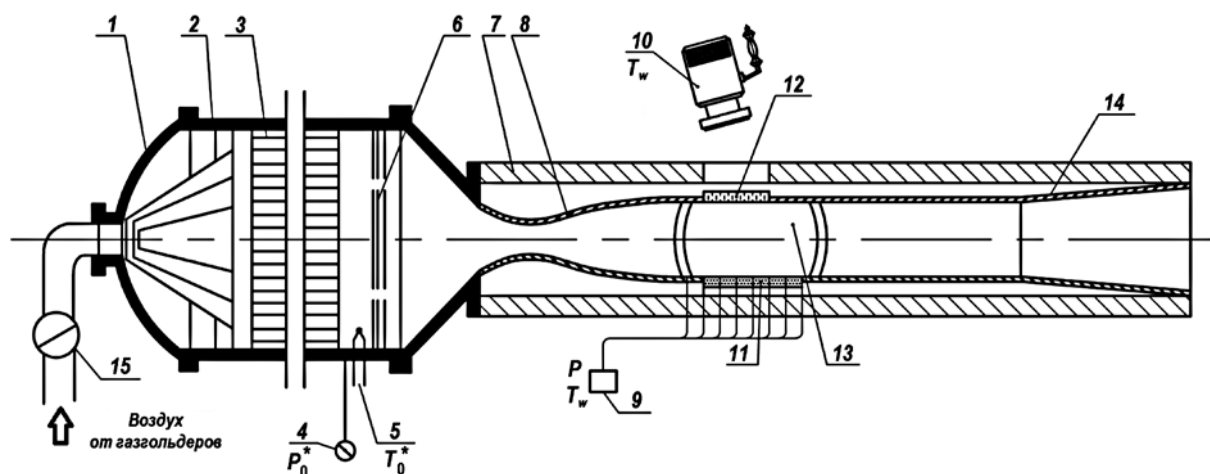


Рис. 2. Общий вид экспериментальной установки:

1 — ресивер; 2 — сборка конусов; 3 — хонейкомб; 4 — манометр; 5 — термопара для измерения температуры торможения; 6 — пакет сеток; 7 — рабочая часть; 8 — плоское регулируемое сопло; 9 — отборы статического давления и каналы термопар; 10 — тепловизор; 11 — модель; 12 — инфракрасный иллюминатор; 13 — иллюминатор из оптического стекла; 14 — диффузор; 15 — регулируемая задвижка

(рис. 2) [37]. Давление торможения в ресивере P_0^* составляло до 7 атм, температура торможения менялась в диапазоне $T_0^* = 278...298$ К. Максимальный расход воздуха через установку составлял до 5 кг/с.

Рабочая часть трубы имеет прямоугольное поперечное сечение с размерами 70×90 мм. На боковых стенках рабочего канала для наблюдения картины течения смонтированы иллюминаторы с оптическими защитными стеклами. Визуализация потока осуществлялась оптическим методом с помощью прибора Теплера (ИАБ-451). Таким образом, была возможность наблюдать картину течения и возникающие системы косых скачков уплотнения в канале.

Экспериментальная модель представляет собой пластину из оргстекла — материала с низким коэффициентом теплопроводности $\lambda = 0,19$ Вт/(м·К). Использование такого материала позволило считать пластину теплоизолированной. Модель устанавливали на нижней стенке рабочей части аэродинамической трубы параллельно основному потоку. Ширина модели соответствовала ширине рабочей части трубы — 70 мм, длина $L = 200$ мм. Толщина пограничного слоя на срезе сопла составляла около 6 мм. Критерий Рейнольдса рассчитывали по длине динамического пограничного слоя (расстоянию от горла сопла), на срезе сопла он составлял не менее $Re_x \geq 2 \cdot 10^7$, что свидетельствует о турбулентном режиме течения.

Для измерения распределения статического давления в пластине был сделан дренаж по центральной линии, через который осуществлялся отбор воздуха к датчикам давления ИКД-27 Да-1500. Параметры торможения потока контролировались в форкамере перед сверхзвуковым соплом: давление торможения — датчиками абсолютного давления ИКД-6 ТДа-7 и Honeywell ML-300PS2PC, температура тор-

можения — двумя независимыми хромель-алюмелевыми термопарами. На поверхности и внутри модели были запрессованы десять термопар для регистрации значения и скорости изменения температуры стенки модели. Холодные спаи термопар помещали в сосуд Дьюара, температуру в котором поддерживали на уровне 0 °С и контролировали с помощью лабораторного электронного термометра ЛТ-300.

2. Система автоматизации эксперимента

В данном экспериментальном исследовании требовалось определять большое число параметров (поля давлений и температур), что привело к необходимости накапливать информацию с различного рода сенсорных устройств. Для этих целей экспериментальный стенд был оборудован аппаратурой, способной работать в многоточечном режиме и отвечать высоким требованиям по надежности и точности регистрируемых данных.

На рис. 3 приведена система сбора данных (DAQ — *data acquisition system*), в состав которой входят датчики и измерительные преобразователи, нормирующие устройства (устройства кондиционирования сигналов), соединительные кабели между различными устройствами и аксессуарами к ним, DAQ-устройство, программное обеспечение и персональный компьютер. DAQ-устройство дискретизирует сигналы, выполняет аналого-цифровые преобразования, а также формирует сигналы цифрового вывода. Данные со всех датчиков стенда собирали в коннекторные блоки термопар и датчиков давления NI SCXI-1303, далее через усилители SCXI-1102 (термопарный) и SCXI-1102B (для датчиков давления) подавали в аналого-цифровой преобразователь NI PCI-6220.

Программа эксперимента написана на языке G-программирования в среде LabVIEW и отобра-



Рис. 3. Схема измерительно-регистрирующей системы

жается в виде виртуальных приборов, позволяющих в режиме реального времени отслеживать параметры на экране монитора персонального компьютера (ПК). Программа позволяет осуществлять накопление наблюдений по мере анализа состояния процесса по показаниям виртуального прибора многократно с фиксированием результатов в протокол. Проведению экспериментов предшествовала тарировка датчиков.

Общий вид окна управления экспериментом в среде LabVIEW представлен на рис. 4. В окне показана только вкладка наблюдений с виртуальными

приборами для контроля таких параметров, как полное давление, температура торможения, число Маха набегающего потока, распределение по длине модели статического давления, числа Маха и температуры стенки. На экран выведены и параметры инициализации, монитор выхода параметров на установившийся режим, кнопка записи информации в файл.

Код программы получения и обработки экспериментальных данных показан на рис. 5 (см. третью и четвертую сторону обложки) в последовательности работы слева — направо. Код программы реализован

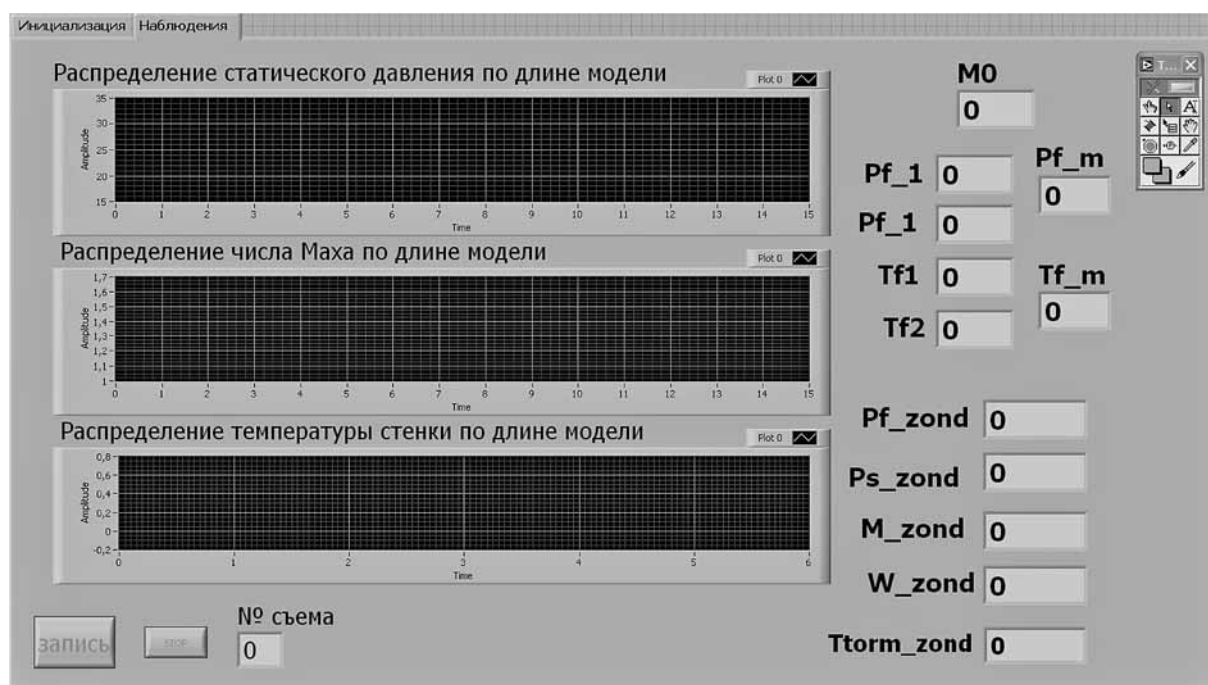


Рис. 4. Общий вид окна управления экспериментом в среде LabVIEW

в виде отдельных фреймов, позволяющих упростить процесс отладки и создания новых кодов путем использования готовых блоков в новых программах.

На рис. 5, а (см. третью сторону обложки) в левой части продемонстрирована инициализация. Показана процедура считывания показаний датчиков давления и термопар при дифференциальном подключении. При этом задаются минимальное и максимальное значения показаний. Задается путь к файлу тарировки. Далее в следующем фрейме формируется протокол наблюдений с перечислением всех записываемых параметров: время съема, давление торможения, 15 показаний статического давления по длине модели, рассчитанные числа Маха в 15 точках, температура торможения, давление и температура в потоке при зондовых измерениях, температура стенки модели в 6 точках и показания 4 термопар, запрессованных на разной глубине модели (для контроля режима теплоизолированной стенки).

На рис. 5, б (см. третью сторону обложки) представлена первичная обработка информации с процедурой перевода показаний напряжения датчиков в значения температуры и давления. При этом используются градуировочные коэффициенты каналов давления (слева) и температур (справа). После чего параметры выводятся на монитор для контроля выхода на установившийся режим (см. рис. 4).

На рис. 5, в (см. четвертую сторону обложки) в левой части представлен расчетный блок, в котором осуществляется первичная обработка наблюдений. В верхней части приведен расчет числа Маха в 15 точках по длине модели, а значение числа Маха набегающего потока выводится в виде виртуального прибора на окно управления экспериментом (см. рис. 4). Далее осуществляется расчет средних показаний параметров торможения в форкамере установки — температуры и давления — по показаниям пары независимых датчиков каждого вида. В последнем блоке данного фрейма представлен расчет параметров в потоке по показаниям зондовых измерений давления и температуры. В правой части рис. 5, в показана функция задания интервала опроса датчиков — 1000 мс. В этом фрейме и далее на рис. 5, г (см. четвертую сторону обложки) последовательно представлены три блока, в которых происходит считывание показаний по измерительным каналам, первичная обработка наблюдений и запись результатов в протокол.

3. Методика обработки экспериментальных данных

Методика проведения исследования заключалась в регистрации изменения параметров на стенке модели от момента запуска аэродинамической трубы в течение 90 с с частотой 1 Гц [31, 36, 37]. Скорость охлаждения пластины отслеживалась с помощью

запрессованных на поверхности модели термопар. Полученные данные позволили восстановить тепловой поток на стенке с помощью интегрирования производной температуры стенки по времени:

$$q = \frac{\sqrt{\rho\lambda C}}{\sqrt{\pi}} \int_0^t \frac{dT_w(\tau)}{(t-\tau)^{1/2}} d\tau, \quad (5)$$

где ρ — плотность материала модели, кг/м³; λ — коэффициент теплопроводности материала модели, Вт/(м·К); C — удельная теплоемкость материала модели, Дж/(кг·К); t — продолжительность эксперимента, с; τ — переменная времени, с.

Выражение (5) справедливо в рамках одномерного уравнения теплопроводности в полубесконечном теле [3]. В процессе проведения эксперимента достижение адиабатной температуры стенки было практически затруднено. Для этого необходимо проведение длительного эксперимента. При этом задача усложняется еще и уменьшением температуры торможения потока в трубе со временем вследствие снижения полного давления в ресивере и невозможности подогрева потока. Поэтому значение адиабатной температуры стенки определяли с помощью экстраполяции линейной аппроксимации графика функции безразмерного теплового потока Ω (от отношения температуры стенки к температуре торможения (T_w/T_0^*)):

$$\Omega = \frac{q}{\rho_0 w_0 C_p T_0^*}. \quad (6)$$

На пересечении с осью абсцисс достигается режим нулевого значения теплового потока, что и соответствует значению адиабатной температуры стенки.

При этом тангенс угла наклона полученного линейного графика позволяет оценить безразмерный коэффициент теплоотдачи — число Стантона — по соотношению (2). Если выразить функцию Ω через число Стантона и отношение температур T_w/T_0^* , то с учетом выражений (2) и (6) можно записать

$$\Omega = St \frac{T_w^*}{T_0^*} - St \frac{T_w}{T_0^*}. \quad (7)$$

На рис. 6 представлена методика определения адиабатной температуры стенки с помощью экстраполяции линейной аппроксимации графика зависимости безразмерного теплового потока (6) от отношения температуры стенки к температуре торможения — до оси абсцисс (нулевого значения теплового потока). Из выражения (7) следует, что график должен быть линейным, причем угол наклона аппроксимирующей прямой соответствует безразмерному коэффициенту

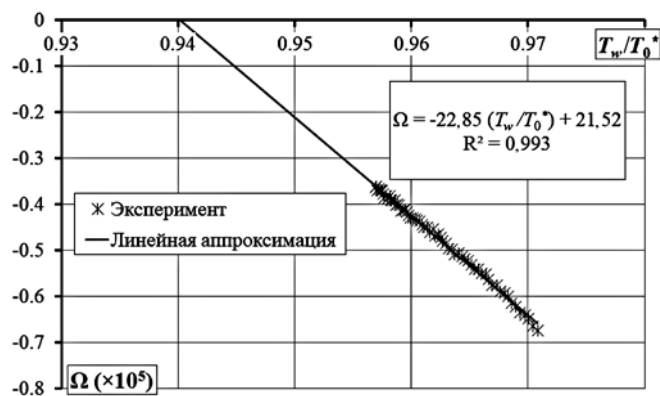


Рис. 6. Методика определения адиабатной температуры стенки по графику зависимости безразмерного теплового потока от отношения температуры стенки к температуре торможения

теплоотдачи — числу Стантона. На рис. 6 показано уравнение аппроксимирующей прямой и величина достоверности линейной аппроксимации R^2 .

Параметры рабочего тела:

$$k := 1.4 \quad R := 287.4 \frac{\text{Дж}}{\text{кг} \cdot \text{К}}$$

Габаритные размеры канала и свойства материала модели:

$$F_1 := 70 \text{ мм} \cdot 90 \text{ мм} = 6.3 \times 10^{-3} \text{ м}^2$$

$$\rho_{\text{мат}} := 1190 \frac{\text{кг}}{\text{м}^3} \quad c_{\text{уд}} := 1367 \frac{\text{Дж}}{\text{кг} \cdot \text{К}} \quad \lambda_{\text{мат}} := 0.2 \frac{\text{Вт}}{\text{м} \cdot \text{К}}$$

Параметры торможения потока в форкамере установки:

$$T_0 := 21^\circ \text{C} = 294.15 \text{ К} \quad P_0 := 604.5 \text{ кПа} = 5.97 \text{ атм}$$

Расчет статических параметров в потоке, плотности, скорости и числа Маха:

$$P_{\text{ст}} := 47.7 \text{ кПа} = 0.471 \text{ атм}$$

$$M := \sqrt{\frac{2}{k-1} \left[\left(\frac{P_0}{P_{\text{ст}}} \right)^{\frac{k-1}{k}} - 1 \right]} = 2.3$$

$$T_0 := \frac{T_0}{1 + \frac{k-1}{2} M^2} = 142.4 \text{ К}$$

$$\rho_0 := \frac{P_{\text{ст}}}{R \cdot T_0} = 1.17 \frac{\text{кг}}{\text{м}^3}$$

$$a_0 := \sqrt{k \cdot R \cdot T_0} = 239.4 \frac{\text{м}}{\text{с}} \quad w_0 := M \cdot a_0 = 552.6 \frac{\text{м}}{\text{с}}$$

Площадь критического сечения сопла:

Given

$$\frac{F_1}{F_{\text{крит}}} = \frac{(1 + 0.2 \cdot M^2)^3}{1.73 \cdot M}$$

$$F_{\text{кр}} := \text{Find}(F_{\text{крит}}) = 2.854 \times 10^{-3} \text{ м}^2$$

Расход воздуха через установку:

$$G_0 := \frac{P_0 \cdot F_{\text{кр}}}{\sqrt{T_0}} \cdot \left(\frac{2}{k+1} \right)^{\frac{k+1}{2(k-1)}} \cdot \sqrt{\frac{k}{R}} = 4.1 \frac{\text{кг}}{\text{с}}$$

Расчет теплового потока в стенку:

$$\tau := 0.1 \dots 89$$

$$q(\tau) := \frac{\sqrt{\rho_{\text{мат}} c_{\text{уд}} \lambda_{\text{мат}}}}{\sqrt{\pi}} \int_0^t \frac{\frac{dT_s(\tau)}{d\tau} \frac{1}{(t-\tau)^2}}{d\tau} d\tau$$

q =	21	22	23	24	25	26	27	28	29	30	31	Вт
	0	-1407	-1386	-1355	-1340	-1326	-1283	-1266	-1234	-1219	-1207	м²

Безразмерный тепловой поток:

$$\Omega(\tau) := \frac{q(\tau \cdot s)}{\rho_0 w_0 C_p T_0 (\tau \cdot s)}$$

$\Omega =$	21	22	23	24	25	26	27	28
0	$-7.4 \cdot 10^{-6}$	$-7.3 \cdot 10^{-6}$	$-7.1 \cdot 10^{-6}$	$-7 \cdot 10^{-6}$	$-7 \cdot 10^{-6}$	$-6.7 \cdot 10^{-6}$	$-6.7 \cdot 10^{-6}$	...

Расчет адиабатной температуры стенки и коэффициента восстановления температуры:

$$f := 0.942$$

$$T_w := f \cdot T_0 = 277.1 \text{ К}$$

Given

$$1 + f \cdot \frac{k-1}{2} M^2 = \frac{T_w}{T_0}$$

$$r := \text{Find}(f) = 0.888$$

Расчет коэффициента теплоотдачи и числа Стантона:

$$\alpha(t) := \frac{q(t)}{T_w(t) - T_w} \quad St(t) := \frac{\alpha(t)}{\rho_0 w_0 C_p}$$

α =	21	22	23	24	25	26	27	28	29	Вт
	0	143.5	143.5	143.6	143.8	143.8	143.8	143.9	143.9	м²К

St =	21	22	23	24	25	26	27	28
	0	2.4·10 ⁻⁴	2.4·10 ⁻⁴	2.4·10 ⁻⁴	2.4·10 ⁻⁴	2.3·10 ⁻⁴	2.3·10 ⁻⁴	...

Рис. 7. Пример обработки экспериментальных данных с расчетом тепловых и динамических параметров в сверхзвуковом потоке

рассчитывается адиабатная температура стенки и коэффициент восстановления температуры. Зная тепловой поток и температурный перепад между температурой стенки и адиабатной температурой, можно определить размерный коэффициент теплоотдачи и безразмерный критерий Стантона по выражению (2).

Оценка неопределенностей термогазодинамических измерений получена для 95 % доверительного интервала по методике, изложенной в работах [38, 39]. Неопределенность измерения безразмерного коэффициента теплоотдачи Стантона $(U_{St})_{0,95}$ составляет около $\pm 9,5\%$ и выражается неопределенностями температуры стенки ΔT_w (по измерениям термодатчиков), температуры торможения ΔT_0^* и давлений ΔP_0^* и ΔP (в меньшей степени):

$$(U_{St})_{0,95} = \sqrt{\left(\frac{\partial St}{\partial T_w}\right)^2 (\Delta T_w)^2 + \left(\frac{\partial St}{\partial T_0^*}\right)^2 (\Delta T_0^*)^2 + \left(\frac{\partial St}{\partial P_0^*}\right)^2 (\Delta P_0^*)^2 + \left(\frac{\partial St}{\partial P}\right)^2 (\Delta P)^2}$$

Неопределенность измерения коэффициента восстановления температуры $(U_r)_{0,95}$ составляет около $\pm 1\%$ и выражается, в первую очередь, неопределенностью температуры стенки ΔT_w в каждый момент времени, при этом наибольшие отклонения от среднего значения аппроксимирующей экспериментальные точки прямой фиксируются при максимальном и минимальном угле наклона данной прямой (число Стантона):

$$(U_r)_{0,95} = \sqrt{\left(\frac{\partial r}{\partial T_w}\right)^2 (\Delta T_w)^2 + \left(\frac{\partial r}{\partial T_0^*}\right)^2 (\Delta T_0^*)^2 + \left(\frac{\partial r}{\partial P_0^*}\right)^2 (\Delta P_0^*)^2 + \left(\frac{\partial r}{\partial P}\right)^2 (\Delta P)^2}$$

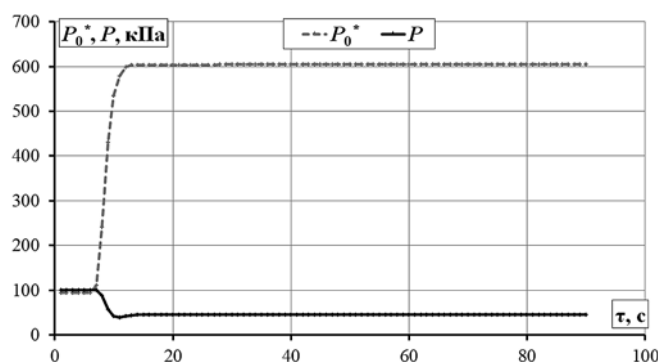


Рис. 8. График изменения полного давления в форкамере P_0^* и статического давления на стенке канала P в процессе проведения эксперимента

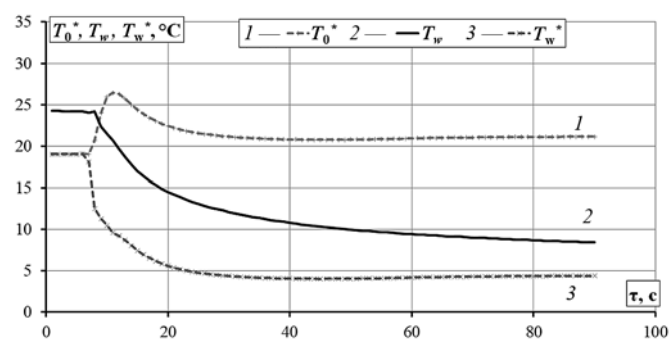


Рис. 9. График изменения температуры торможения в форкамере T_0^* , температуры стенки канала T_w и адиабатной температуры стенки T_w^* в процессе проведения эксперимента

4. Обсуждение результатов

На рис. 8 представлен график изменения давления торможения и статического давления на боковой стенке перед моделью в процессе запуска аэродинамической трубы. После открытия задвижки запуск происходит в течение примерно 4 с, после чего давление в потоке устанавливается и практически не изменяется в процессе проведения эксперимента. Соотношение полного и статического давлений позволяет определить число Маха набегающего потока — около 2,3. Все исследования в данной работе проводили при одинаковом числе Маха на срезе сопла.

На рис. 9 представлен график изменения температуры торможения и температуры стенки в процессе запуска аэродинамической трубы. Как видно, температура торможения в потоке устанавливается медленнее, чем давление (см. рис. 8), но также сохраняется практически постоянной в процессе проведения исследования. Стенка модели при этом охлаждается и в пределе может достигнуть значения адиабатной температуры, однако в данном эксперименте состояния теплового равновесия не достигается за время проведения исследования.

По темпу охлаждения модели с помощью интегрирования выражения (5) удастся построить график изменения теплового потока (рис. 10) за время за-

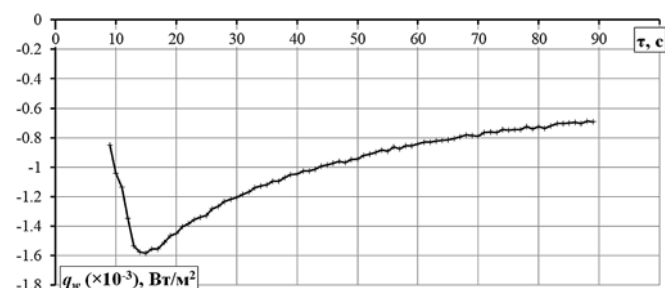


Рис. 10. График изменения теплового потока на стенке модели q_w в процессе проведения эксперимента

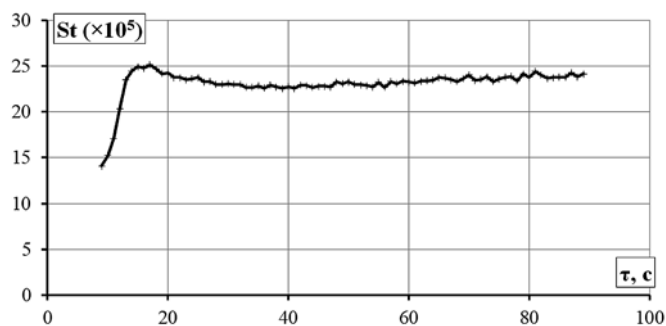


Рис. 11. График изменения числа Стантона St в процессе проведения эксперимента

пуска и проведения эксперимента. Значение теплового потока оказывается отрицательным, поскольку теплота передается от стенки в пограничный слой. Сначала тепловой поток максимальный по модулю и постепенно уменьшается по мере приближения температуры стенки к температуре адиабатически заторможенного газа вблизи стенки.

Как видно на рис. 11, коэффициент теплоотдачи в процессе эксперимента остается практически постоянным. В точке пересечения представленного на рис. 6 графика с осью абсцисс достигается режим нулевого теплового потока, и температура стенки становится равной адиабатной температуре стенки, т. е. температуре адиабатически заторможенного газа в пристенной области.

Заключение

Представлена методика и программное обеспечение для проведения и обработки результатов автоматизированного теплофизического эксперимента на сверхзвуковой аэродинамической установке. Используемые программные средства, разработанные в среде LabVIEW, позволяют в режиме реального времени отслеживать параметры на экране монитора, проводить накопление наблюдений по мере анализа состояния процесса по показаниям виртуального прибора многократно с фиксацией результатов в протокол. Представлена методика и программы для обработки первичных экспериментальных данных с последующим расчетом тепловых и динамических характеристик сверхзвукового потока. В результате исследования получены графики изменения в процессе запуска и работы аэродинамической установки полного и статического давления, температуры торможения, температуры стенки, адиабатной температуры, теплового потока и критерия теплоотдачи Стантона.

Работа выполнена за счет гранта Российского научного фонда (проект № 14-19-00699).

Список литературы

1. Стронгин М. М., Виноградов Ю. А., Здитовец А. Г., Киселёв Н. А., Попович С. С. Использование технологий National Instruments в экспериментальных исследованиях процессов термогазодинамики // Программная инженерия. 2017. Т. 8, № 5. С. 230—240. DOI: 10.17587/prin.8.230-240.
2. Hayes J. R., Neumann R. D. Introduction to the aerodynamic heating analysis of supersonic missiles // Tactical Missile Aerodynamics. Progress in Astronautics and Aeronautics Series. 1992. Vol. 142. P. 63—114.
3. Schultz D. L., Jones T. V. Heat-transfer measurements in short-duration hypersonic facilities. AGARDograph 165, London, 1973. 149 p.
4. Кутателадзе С. С., Леонтьев А. И. Тепломассообмен и трение в турбулентном пограничном слое. М.: Энергоатомиздат, 1985. 320 с.
5. Moffat R. J. What's new in convective heat transfer? // International Journal of Heat and Fluid Flow. 1998. Vol. 19. P. 90—101. DOI: 10.1016/S0142-727X(97)10014-5.
6. Neumann R. D., Freeman D. C. Experimental Measurement of Aerodynamic Heating About Complex Shapes at Supersonic Mach Numbers // J. of Spacecraft and Rockets. 2012. Vol. 49, N. 6. P. 1080—1087.
7. Gomes R. A., Niehuis R. The Concept of Adiabatic Heat Transfer Coefficient and Its Application to Turbomachinery // Proc. of ASME Turbo Expo. 2013. P. 1—8.
8. Lavagnoli S., Maesschalck C. D., Paniagua G. Uncertainty analysis of adiabatic wall temperature measurements in turbine experiments // Applied Thermal Engineering. 2015. Vol. 82. P. 170—181.
9. Лапин Ю. В. Турбулентный пограничный слой в сверхзвуковых потоках газа. М.: Наука 1970, 344 с.
10. Здитовец А. Г. Изучение влияния формы поверхности и критерия Прандтля на температуру теплоизолированного тела, обтекаемого сверхзвуковым потоком газа: дис. ... канд. техн. наук. М., 2005. 108 с.
11. Виноградов Ю. А., Ермолаев И. К., Здитовец А. Г., Леонтьев А. И. Измерение равновесной температуры стенки сверхзвукового сопла при течении смеси газов с низким значением числа Прандтля // Известия Российской академии наук. Энергетика. 2005. № 4. С. 128—133.
12. Леонтьев А. И., Лущик В. Г., Якубенко А. Е. Коэффициент восстановления в сверхзвуковом потоке газа с малым числом Прандтля // Теплофизика высоких температур. 2006. Т. 44, № 2. С. 238—245.
13. Вигдорович И. И., Леонтьев А. И. Энергоразделение газов с малыми и большими числами Прандтля // Известия Российской академии наук. Механика жидкости и газа. 2013. № 6. С. 117—134.
14. Бурцев С. А., Кочуров Д. С., Щеголев Н. Л. Исследование влияния доли гелия на значение критерия Прандтля газовых смесей // Машиностроение и компьютерные технологии. 2014. № 5. С. 314—329. DOI: 10.7463/0514.0710811.

15. **Егоров К. С., Рогожинский К. С.** Численное моделирование влияния числа Прандтля газа и схемы течения на эффективность работы устройства безмашинного энергоразделения // *Машиностроение и компьютерные технологии*. 2015. № 10. С. 21–35.
16. **Лущик В. Г., Макарова М. С.** Численное исследование влияния числа Прандтля на коэффициенты восстановления температуры и аналогии Рейнольдса в пограничном слое на пластине // *Теплофизика высоких температур*. 2016. Т. 54. № 3. С. 401–407.
17. **Леонтьев А. И., Лущик В. Г., Якубенко А. Е.** Сжимаемый турбулентный пограничный слой на проницаемой пластине при вдуве инородного газа // *Теплофизика высоких температур*. 2007. Т. 45, № 4. С. 543–551.
18. **Виноградов Ю. А., Егоров К. С., Попович С. С., Стронгин М. М.** Исследование тепломассообмена на проницаемой поверхности в сверхзвуковом пограничном слое // *Тепловые процессы в технике*. 2010. Т. 1, № 1. С. 7–9.
19. **Виноградов Ю. А., Здитовец А. Г., Медвецкая Н. В., Стронгин М. М., Титов А. А.** Экспериментальное исследование влияния вдува инородного газа в сверхзвуковой поток аргона на температуру адиабатической стенки // *Тепловые процессы в технике*. 2010. № 4. С. 159–163.
20. **Леонтьев А. И., Лущик В. Г., Якубенко А. Е.** Влияние вдува (отсоса) на энергоразделение потоков сжимаемого газа // *Известия Российской академии наук. Механика жидкости и газа*. 2011. № 6. С. 110–117.
21. **Леонтьев А. И., Лущик В. Г., Макарова М. С.** Коэффициент восстановления температуры в пограничном слое на проницаемой пластине // *Теплофизика высоких температур*. 2017. Т. 55, № 2. С. 255–261.
22. **Eckert E. R. G.** Energy separation in fluid streams // *Int. Comm. Heat mass transfer*. 1986. Vol. 13. P. 127–143.
23. **Sanitjia S., Goldstein R. J.** Forced Convection Heat Transfer from a Circular Cylinder in Crossflow to Air and Liquids // *Int. J. Heat Mass Transfer*. 2004. Vol. 47. P. 4795–4805.
24. **Здитовец А. Г., Титов А. А.** Влияние формы поверхности теплоизолированного стержня, омываемого сверхзвуковым потоком, на коэффициент восстановления температуры // *Известия РАН. Энергетика*. 2007. № 2. С. 111–117.
25. **Виноградов Ю. А., Попович С. С., Стронгин М. М.** Экспериментальное исследование влияния отрывного сверхзвукового течения на температуру теплоизолированной стенки // *Труды Шестой Российской национальной конференции по теплообмену*. М.: Изд-во МЭИ, 2014. С. 262–265.
26. **Popovich S. S., Egorov K. S., Vinogradov Yu. A.** Experimental research of adiabatic wall temperature influenced by separated supersonic flow // *Proceedings of 15th International Heat Transfer Conference, IHTC-15-8962*, 2014. P. 1–11.
27. **Попович С. С., Виноградов Ю. А., Стронгин М. М.** Экспериментальное исследование возможности интенсификации теплообмена в устройстве безмашинного энергоразделения потоков // *Вестник СГАУ*. 2015. Т. 14, № 2. С. 159–169. DOI: 10.18287/2412-7329-2015-14-2-159-169.
28. **Popovich S., Strongin M., Vinogradov Yu., Zditovets A.** Experimental research of heat transfer augmentation technique in gas dynamic energy separation process // *Eight International Symposium on turbulence, heat and mass transfer. Proceedings*. 2015. Begell House Inc., 2015. P. 865–868.
29. **Чжен П.** Отрывные течения. Т. 3. М.: Мир, 1973. 335 с.
30. **Gnedin M., Knight D., Zheltovodov A., Maksimov A., Shevchenko A., Vorontsov S.** 3-D crossing shock wave-turbulent boundary layer interaction // *AIAA Paper 96-2001*, 1996. P. 1–18.
31. **Mee D. J., Chiu H. S., Ireland P. T.** Technique for detailed heat transfer measurements in cold supersonic blowdown tunnels using thermochromic liquid crystals // *International Journal of Heat and Mass Transfer*. 2002. Vol. 45. P. 3287–3297.
32. **Smulsky Y. I., Terekhov V. I., Yarygina N. I.** Heat transfer in turbulent separated flow behind a rib on the surface of square channel at different orientation angles relative to flow direction // *International Journal of Heat and Mass Transfer*. 2012. Т. 55, № 4. С. 726–733.
33. **Попович С. С.** Экспериментальное исследование влияния падающего скачка уплотнения на адиабатную температуру стенки в сверхзвуковом потоке сжимаемого газа // *Тепловые процессы в технике*. 2014. Т. 6, № 3. С. 98–104.
34. **Виноградов Ю. А., Здитовец А. Г., Попович С. С., Стронгин М. М., Леонтьев А. И.** Исследование воздействия ударных волн на температурный перепад в устройстве безмашинного энергоразделения потоков // *Будущее машиностроения России. Сборник докладов Восьмой Всероссийской конференции молодых ученых и специалистов*. 2015. С. 511–515.
35. **Попович С. С.** Экспериментальное исследование влияния ударных волн на эффект безмашинного энергоразделения газовых потоков // *Машиностроение и компьютерные технологии*. 2016. № 3. С. 64–80. DOI: 10.7463/0316.0835444.
36. **Виноградов Ю. А., Попович С. С., Стронгин М. М.** Экспериментальное исследование коэффициента восстановления температуры и интенсификации теплоотдачи при течении на плоской стенке за ребром сверхзвукового потока сжимаемого газа // *Машиностроение и компьютерные технологии*. 2016. № 11. С. 55–75. DOI: 10.7463/1116.0849557.
37. **Попович С. С.** Влияние ударных волн на эффект безмашинного энергоразделения: дис. ... канд. техн. наук. М.: НИИ механики МГУ, 2016. 172 с.
38. **Moffat R. J.** Describing the uncertainties in experimental results // *Experimental and Fluid Science*. 1988. P. 3–17.
39. **РМГ 43-2001 ГСИ.** Применение "Руководства по выражению неопределенности измерений". ВНИИМ им. Д. И. Менделеева. 2003. 21 с.

Experimental Automation and Data Processing Features for Supersonic Heat Transfer Research

S. S. Popovich, pss1@mail.ru, Lomonosov Moscow State University, Institute of mechanics, Moscow, 119192, Russian Federation

Corresponding author:

Popovich Sergey S., Research Scientist, Lomonosov Moscow State University, Institute of mechanics, Moscow, 119192, Russian Federation
E-mail: pss1@mail.ru

Received on October 10, 2017

Accepted on November 9, 2017

The paper presents an automation technique for thermophysical experiment in a supersonic wind tunnel by means of LabVIEW automation program and a technique for experimental data processing. The study of the heat transfer parameters was carried out on unsteady state regime during the start-up of the wind tunnel before reaching thermal equilibrium state. Mach number of the oncoming flow was 2.25, Reynolds number based on the length of the dynamic boundary layer was not less than 20 million at the nozzle cutoff. The graphs are presented for the change during the experiment of stagnation pressure and temperature, model wall temperature, static pressure, adiabatic wall temperature, heat flux and Stanton number. Research is carried out on the experimental facilities of the Institute of Mechanics of Lomonosov Moscow State University.

Keywords: LabVIEW, wind tunnel, virtual device, supersonic flow, adiabatic wall temperature, temperature recovery factor, heat transfer coefficient

Acknowledgements: The research was supported by Russian Science Foundation (Project 14-19-00699).

For citation:

Popovich S. S. Experimental Automation and Data Processing Features for Supersonic Heat Transfer Research, *Programnaya Ingeneria*, 2018, vol. 9, no. 1, pp. 35–45.

DOI: 10.17587/prin.9.35-45

References

1. Strongin M. M., Vinogradov Yu. A., Zditovets A. G., Kiselev N. A., Popovich S. S. Ispol'zovaniye tekhnologiy National Instruments v eksperimental'nykh issledovaniyakh protsessov termozodnamiki (Applying of National Instruments Technologies in Experimental Research of Thermal Gas Dynamics Processes), *Programnaya ingeneria*, 2017, vol. 8, no. 5, pp. 230–240 DOI: 10.17587/prin.8.230-240 (in Russian).
2. Hayes J. R., Neumann R. D. Introduction to the aerodynamic heating analysis of supersonic missiles, *Tactical Missile Aerodynamics. Progress in Astronautics and Aeronautics Series*, 1992, vol. 142, pp. 63–114.
3. Schultz D. L., Jones T. V. Heat-transfer measurements in short-duration hypersonic facilities. AGARDograph 165, London, 1973, 149 p.
4. Kutateladze S. S., Leontiev A. I. Teplomassobmen i trenie v turbulentnom pogranchnom sloe (Heat Mass transfer and friction in turbulent boundary layer), Moscow, Energoatom Publ., 1985, 320 p. (in Russian).
5. Moffat R. J. What's new in convective heat transfer? *International Journal of Heat and Fluid Flow*, 1998, vol. 19, pp. 90–101. DOI: 10.1016/S0142-727X(97)10014-5.
6. Neumann R. D., Freeman D. C. Experimental Measurement of Aerodynamic Heating About Complex Shapes at Supersonic Mach Numbers, *J. of Spacecraft and Rockets*, 2012, vol. 49, no. 6, pp. 1080–1087.
7. Gomes R. A., Niehuis R. The Concept of Adiabatic Heat Transfer Coefficient and Its Application to Turbomachinery, *Proc. of ASME Turbo Expo*, 2013, pp. 1–8.
8. Lavagnoli S., Maesschalck C. D., Paniagua G. Uncertainty analysis of adiabatic wall temperature measurements in turbine experiments, *Applied Thermal Engineering*, 2015, vol. 82, pp. 170–181.
9. Lapin Yu. V. *Turbulentnyy pogranchnyy sloy v sverkhzvukovykh potokakh gaza* (Turbulent boundary layer in supersonic gas flows), Moscow, Nauka, 1970, 344 p. (in Russian).
10. Zditovets A. G. Izucheniye vliyaniya formy poverkhnosti i kriteriya Prandtlya na temperaturu teploizolirovannogo tela, obtekayemogo sverkhzvukovym potokom gaza (Study of the influence of the surface shape and Prandtl number on the temperature of a thermally insulated body flowing in a supersonic gas flow): dis. ... kand. tekhn. nauk. Moscow, 2005, 108 p. (in Russian).
11. Vinogradov Yu. A., Yermolayev I. K., Zditovets A. G., Leont'yev A. I. Izmereniye ravnovesnoy temperatury stenki sverkhzvukovogo sopla pri techenii smesi gazov s nizkim znacheniyem chisla Prandtlya (Measurement of the equilibrium wall temperature of a supersonic nozzle in the flow of a mixture of gases with a low value of Prandtl number), *Izvestiya Rossiyskoy akademii nauk. Energetika*, 2005, no. 4, pp. 128–133 (in Russian).
12. Leont'yev A. I., Lushchik V. G., Yakubenko A. Ye. Koefitsiyent vosstanovleniya v sverkhzvukovom potoke gaza s malym chislom Prandtlya (Recovery factor in a supersonic gas flow with low Prandtl number), *Teplofizika vysokikh temperature*, 2006, vol. 44, no. 2, pp. 238–245 (in Russian).

13. **Vigdorovich I. I., Leont'yev A. I.** Energorazdeleniye gazov s malymi i bol'shimi chislami Prandtl'ya (Energy separation of gases with low and high Prandtl numbers), *Izvestiya Rossiyskoy akademii nauk. Mekhanika zhidkosti i gaza*, 2013, no. 6, pp. 117–134 (in Russian) DOI: 10.7463/0514.2719811.
14. **Burtsev S. A., Kochurov D. S., Shchegolev N. L.** Issledovaniye vliyaniya doli geliya na znacheniya kriteriya Prandtl'ya gazovykh smesey (Investigation of the effect of the proportion of helium on the value of Prandtl number for gas mixtures), *Mashinostroyeniye i komp'yuternyye tekhnologii*, 2014, no. 5, pp. 314–329 DOI: 10.7463/0514/0710811 (in Russian).
15. **Egorov K. S., Rogozhinskiy K. S.** Chislennoye modelirovaniye vliyaniya chisla Prandtl'ya gaza i skhemy techeniya na effektivnost' raboty ustroystva bezmashinnogo energorazdeleniya (Numerical simulation of the influence of gas Prandtl number and the flow scheme on the efficiency of a machine-free energy separation device), *Mashinostroyeniye i komp'yuternyye tekhnologii*, 2015, no. 10, pp. 21–35 (in Russian).
16. **Lushchik V. G., Makarova M. S.** Chislennoye issledovaniye vliyaniya chisla Prandtl'ya na koeffitsiyenty vosstanovleniya temperatury i analogii Reynol'dsa v pogranichnom sloye na plastine (Numerical investigation of the influence of Prandtl number on the temperature recovery coefficients and Reynolds analogy in the boundary layer on the plate), *Teplofizika vysokikh temperatur*, 2016, vol. 54, no. 3, pp. 401–407 (in Russian).
17. **Leont'yev A. I., Lushchik V. G., Yakubenko A. Ye.** Szhimayemyy turbulentnyy pogranichnyy sloy na pronitsayemoy plastine pri vduve inorodnogo gaza (Compressible turbulent boundary layer on a permeable plate during injection of a foreign gas), *Teplofizika vysokikh temperatur*, 2007, vol. 45, no. 4, pp. 543–551 (in Russian).
18. **Vinogradov Yu. A., Egorov K. S., Popovich S. S., Strongin M. M.** Issledovaniye teplomassoobmena na pronitsayemoy poverkhnosti v sverkhzvukovom pogranichnom sloye (Investigation of heat and mass transfer on a permeable surface in a supersonic boundary layer), *Teplovyye protsessy v tekhnike*, 2010, vol. 1, no. 1, pp. 7–9 (in Russian).
19. **Vinogradov Yu. A., Zditovets A. G., Medvetskaya N. V., Strongin M. M., Titov A. A.** Eksperimental'noye issledovaniye vliyaniya vduva inorodnogo gaza v sverkhzvukovoy potok argona na temperaturu adiabaticheskoy stenki (Experimental study of the effect of injection of a foreign gas into a supersonic argon flow on adiabatic wall temperature), *Teplovyye protsessy v tekhnike*, 2010, no. 4, pp. 159–163 (in Russian).
20. **Leont'yev A. I., Lushchik V. G., Yakubenko A. Ye.** Vliyaniye vduva (otsosa) na energorazdeleniye potokov szhimayemogo gaza (Influence of blowing (suction) on the energy separation of compressible gas flows), *Izvestiya Rossiyskoy akademii nauk. Mekhanika zhidkosti i gaza*, 2011, no. 6, pp. 110–117 (in Russian).
21. **Leont'yev A. I., Lushchik V. G., Makarova M. S.** Koeffitsiyent vosstanovleniya temperatury v pogranichnom sloye na pronitsayemoy plastine (Temperature recovery factor in the boundary layer on a permeable plate), *Teplofizika vysokikh temperatur*, 2017, vol. 55, no. 2, pp. 255–261 (in Russian).
22. **Eckert E. R. G.** Energy separation in fluid streams, *Int. Comm. Heat mass transfer*, 1986, vol. 13, pp. 127–143.
23. **Sanitja S., Goldstein R. J.** Forced Convection Heat Transfer from a Circular Cylinder in Crossflow to Air and Liquids, *Int. J. Heat Mass Transfer*, 2004, vol. 47, pp. 4795–4805.
24. **Zditovets A. G., Titov A. A.** Vliyaniye formy poverkhnosti teploizolirovannogo sterzhnya, omyvayemogo sverkhzvukovym potokom, na koeffitsiyent vosstanovleniya temperatury (Influence of the surface shape of a heat-insulated rod washed by a supersonic flow on a temperature recovery factor), *Izvestiya RAN. Energetika*, 2007, no. 2, pp. 111–117 (in Russian).
25. **Vinogradov Yu. A., Popovich S. S., Strongin M. M.** Eksperimental'noye issledovaniye vliyaniya otryvnogo sverkhzvukovogo techeniya na temperaturu teploizolirovannoy stenki (Experimental study of supersonic separated flow influence on heat-insulated wall temperature), *Trudy Shestoy Rossiyskoy natsional'noy konferentsii po teploobmenu*, Moscow, Izd-vo MEI, 2014, pp. 262–265 (in Russian).
26. **Popovich S. S., Egorov K. S., Vinogradov Yu. A.** Experimental research of adiabatic wall temperature influenced by separated supersonic flow, *Proceedings of 15th International Heat Transfer Conference, IHTC-15-8962*, 2014, pp. 1–11.
27. **Popovich S. S., Vinogradov Yu. A., Strongin M. M.** Eksperimental'noye issledovaniye vozmozhnosti intensivatsii teploobmena v ustroystve bezmashinnogo energorazdeleniya potokov (Experimental research of heat transfer augmentation in gas dynamic energy separation process), *Vestnik SGAU*, 2015, vol. 14, no. 2, pp. 159–169. DOI: 10.18287/2412-7329-2015-14-2-159-169 (in Russian).
28. **Popovich S., Strongin M., Vinogradov Yu., Zditovets A.** Experimental research of heat transfer augmentation technique in gas dynamic energy separation process, *Eight international symposium on turbulence, heat and mass transfer. Proceedings*, 2015, Begell House Inc. 2015, pp. 865–868.
29. **Chzhen P.** Otryvnyye techeniya (Separated Flows), Moscow, Mir, vol. 3, 1973, 335 p. (in Russian).
30. **Gnedin M., Knight D., Zheltovodov A., Maksimov A., Shevchenko A., Vorontsov S.** 3-D crossing shock wave-turbulent boundary layer interaction, *AIAA Paper 96-2001*, 1996, p. 1–18.
31. **Mee D. J., Chiu H. S., Ireland P. T.** Technique for detailed heat transfer measurements in cold supersonic blowdown tunnels using thermochromic liquid crystals, *International Journal of Heat and Mass Transfer*, 2002, vol. 45, pp. 3287–3297.
32. **Smulsky Y. I., Terekhov V. I., Yarygina N. I.** Heat transfer in turbulent separated flow behind a rib on the surface of square channel at different orientation angles relative to flow direction, *International Journal of Heat and Mass Transfer*, 2012, vol. 55, no. 4, pp. 726–733.
33. **Popovich S. S.** Eksperimental'noye issledovaniye vliyaniya padayushchego skachka uplotneniya na adiabatnyuyu temperaturu stenki v sverkhzvukovom potoke szhimayemogo gaza (Experimental study of an incident shock wave influence on adiabatic wall temperature in a supersonic compressible gas flow), *Teplovyye protsessy v tekhnike*, 2014, vol. 6, no. 3, pp. 98–104 (in Russian).
34. **Vinogradov Yu. A., Zditovets A. G., Popovich S. S., Strongin M. M., Leont'yev A. I.** Issledovaniye vozdeystviya udarnykh voln na temperaturnyy perepad v ustroystve bezmashinnogo energorazdeleniya potokov (Investigation of the impact of shock waves on the temperature difference in a machine-free energy separation device), *Budushcheye mashinostroyeniya Rossii. Sbornik dokladov Vos'moy Vserossiyskoy konferentsii molodykh uchenykh i spetsialistov*, 2015, pp. 511–515 (in Russian).
35. **Popovich S. S.** Eksperimental'noye issledovaniye vliyaniya udarnykh voln na effekt bezmashinnogo energorazdeleniya gazovykh potokov (Experimental research of machineless energy separation effect influenced by shock waves), *Mashinostroyeniye i komp'yuternyye tekhnologii*, 2016, no. 3, pp. 64–80. DOI: 10.7463/0316.0835444 (in Russian).
36. **Vinogradov Yu. A., Popovich S. S., Strongin M. M.** Eksperimental'noye issledovaniye koeffitsiyenta vosstanovleniya temperatury i intensivatsii teplootdachi pri techenii na ploskoy stenke za rebrom sverkhzvukovogo potoka szhimayemogo gaza (Experimentally studied recovery factor and heat transfer augmentation with supersonic flow on a flat wall behind the edge), *Mashinostroyeniye i komp'yuternyye tekhnologii*, 2016, no. 11, pp. 55–75. DOI: 10.7463/1116.0849557 (in Russian).
37. **Popovich S. S.** Vliyaniye udarnykh voln na effekt bezmashinnogo energorazdeleniya (Shock wave influence on energy separation effect): dis. kand. tekhn. nauk, Moscow, NII mekhaniki MGU, 2016, 172 p. (in Russian).
38. **Moffat R. J.** Describing the uncertainties in experimental results, *Experimental and Fluid Science*, 1988, pp. 3–17.
39. **RMG 43-2001 GSI.** Primeneniye "Rukovodstva po vyrashcheniyu neopredelennosti izmereniy" (Application of the "Guide to the expression of measurement uncertainty"), VNIIM im. D. I. Mendeleeva, 2003, 21 p. (in Russian).

Итоги тринадцатой конференции "Разработка ПО/СЕЕ-SECR 2017"

Октябрь этого года профессионалам ИТ-индустрии запомнился прошедшей конференцией "SECR 2017/Разработка ПО", ведь впервые за тринадцатилетнюю историю это ежегодное событие прошло в Санкт-Петербурге.

Около 600 участников из разных стран — специалистов и менеджеров ИТ-компаний, научных деятелей, предпринимателей, студентов и инвесто-

ров собрал на одной площадке интерес к последним трендам в разработке ПО. Участники особенно отметили атмосферу, которая способствовала новым знакомствам и эффективному нетворкингу.

Software Engineering Conference Russia включала два насыщенных дня докладов и дополнительный день, полностью посвященный мастер-классам, где гости оттачивали новые техники в сфере разра-



Председатель конференции Николай Пунтиков



Эрик Райс

ботки ПО. Программа основных дней была сформирована таким образом, что в четырех залах шли презентации и панельные дискуссии, а один зал был отведен для воркшопов. Особенность конференции, которую отмечают организаторы, — большой тематический охват. Были охвачены и узкоспециализированные технические темы, и общие вопросы, например, о руководстве командой.

Помимо спикеров, доклады которых прошли жесткий конкурсный отбор, на мероприятии присутствовали приглашенные докладчики, среди которых Андрей Николаевич Терехов — профессор, заведующий кафедрой системного программирования СПбГУ, Ивар

Яacobсон — основоположник программной инженерии в областях сценариев использования, UML, Эрик Райс — автор книги "Usable Usability", Асхат Уразбаев (ScrumTrek), Анатолий Шкред (ИНТУИТ.РУ).

"Мы приглашаем и поощряем тех, кто выступает с научными докладами", — говорит председатель конференции Николай Пунтиков. Так, на SECR уже не первый год вручается премия имени Бертрана Мейера за лучшую исследовательскую работу в области программной инженерии. В этом году победителями стали сразу два автора: Николай Ефанов, МФТИ (статья "Построение формальной грамматики системных вызовов")



Ивар Яacobсон



Асхат Уразбаев



Андрей Николаевич Терехов



Мастер-класс

и Светлана Лазарева, Рэйдикс (статья "Умный "фейс-контроль": алгоритмы машинного обучения для эффективного кэширования данных на SSD").

Конференцию SECR 2017 поддержали такие компании, как Jet Brains, First Line Software, Dell EMC, DINS, Veeam, а также Auriga, T-Systems,

Айти, Reksoft, Единая Фронтальная Система, лично Андрей Николаевич Терехов, ассоциация РУССОФТ и др.

Видеозаписи и презентации докладов Software Engineering Conference Russia доступны на сайте www.secr.ru.

ООО "Издательство "Новые технологии". 107076, Москва, Стромынский пер., 4
Технический редактор *Е. М. Патрушева*. Корректор *Е. В. Комиссарова*

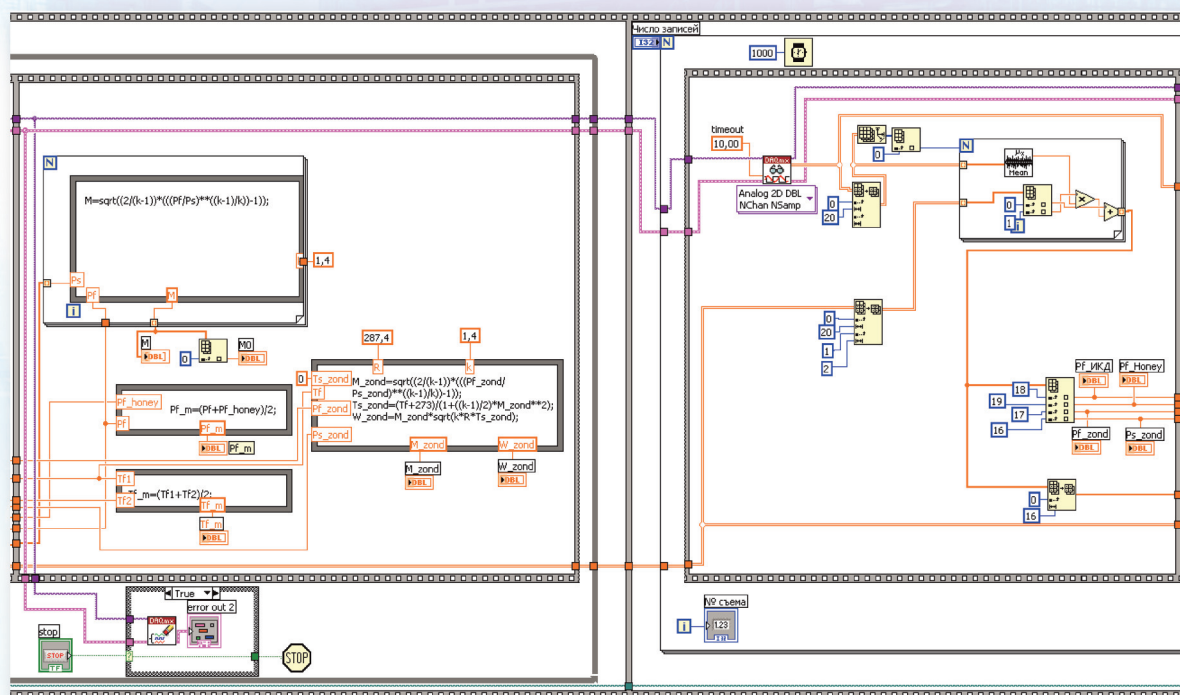
Сдано в набор 17.11.2017 г. Подписано в печать 19.12.2017 г. Формат 60×88 1/8. Заказ РИ18
Цена свободная.

Оригинал-макет ООО "Авансд солюшнз". Отпечатано в ООО "Авансд солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru

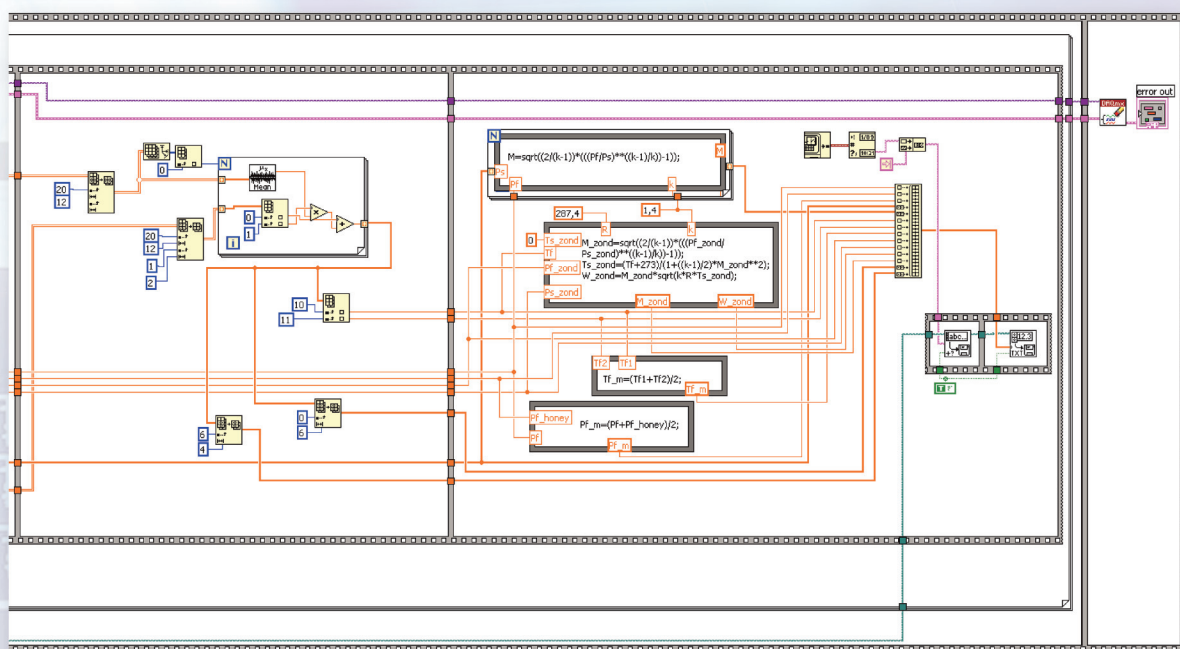
[illegible]

Рис. 5. Код программы получения и обработки экспериментальных данных в среде LabVIEW

Рисунок к статье С. С. Поповича
 «ОСОБЕННОСТИ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТА
 И ОБРАБОТКИ РЕЗУЛЬТАТОВ ПРИ ИССЛЕДОВАНИИ ТЕПЛООБМЕНА
 В СВЕРХЗВУКОВОМ ПОТОКЕ СЖИМАЕМОГО ГАЗА»



б)



2)

Рис. 5. Код программы получения и обработки экспериментальных данных в среде LabVIEW