

Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

ГОРЕЛОВ Алексей Анатольевич

**Анализ задержек в коммуникационной среде
вычислительного кластера**

ДИПЛОМНАЯ РАБОТА

Научный руководитель:

к.ф-м.н.

Майсурадзе Арчил Ивериевич

Москва, 2015

Содержание

1	Введение	4
2	Анализ индивидуальных задержек	6
2.1	Модель вычислительного кластера	6
2.2	Модель задержек	9
2.3	Описание данных	10
2.4	Технология сбора данных	12
2.5	Обзор моделей распределений задержек	13
2.6	Методы оценки параметров модели	14
2.6.1	Постановка задачи	14
2.6.2	EM-алгоритм	14
2.6.3	Метод минимизации расстояния	15
2.7	Тестирование методов оценки параметров модели	17
2.7.1	Предобработка данных	17
2.7.2	Результаты тестирования	17
2.8	Выводы	18
3	Анализ задержек на маршрутах	23
3.1	Свойства монотонности, неравенства треугольника и неделимости	23
3.2	Проверка свойств на полученных данных	25
3.2.1	Проверка свойств на выборках	26
3.2.2	Проверка свойств на моделях	27
3.2.3	Сравнение результатов проверок свойств на моделях и на данных	28
3.3	Выводы	29

4	Изучение зависимостей параметров моделей от характеристик сообщений	31
4.1	Кластеризация пар узлов на основе данных о задержках	31
4.1.1	Описание метода кластеризации	33
4.1.2	Тестирование метода	34
4.2	Выводы	35
5	Заключение	40
	Список литературы	40

Глава 1

Введение

При распараллеливании сложных задач на вычислительном кластере часто встаёт проблема планирования порядка выполнения подзадач исходной задачи. Если раньше для этого применялись методы статического составления расписания, то есть планирование проводилось до запуска задачи на кластере, то в настоящий момент всё чаще используются динамические методы, выполняющие планирование в ходе выполнения задачи. При этом для оптимизации общего времени решения задачи на кластере необходимо учесть множество факторов, специфичных для каждого вычислительного кластера, в том числе и задержки пересылки данных от одного узла к другому.

Отметим, что причины появления этих задержек многочисленны и специфичны для каждой системы, поэтому построение моделей задержек на основании изучения факторов их возникновения предполагает подробное изучение программного и аппаратного обеспечения на каждом уровне рассматриваемого сетевого протокола, что возможно только для конкретных небольших систем и конкретных задач на них.

В данной работе предлагается иной подход, предполагающий абстрагирование от конкретной реализации сетевого взаимодействия между процессами. Утверждается, что возможно построить адекватную модель задержек, которую можно использовать, например, в задачах динамического планирования расписания или в задачах диагностики кластера, отталкиваясь только от «сырых данных», полученных в результате тестирования сети специальными программными системами. Для этого неконтролируемые факторы выносятся в область скрытых переменных модели, а са-

ми задержки рассматриваются как случайные величины с некоторым неизвестными нам распределениями.

Для сбора информации о задержках обычно используются специальные пакеты программного обеспечения. Часто результатом их работы являются некоторые статистики полученной в ходе тестов выборки задержек. Например, на основании выборки задержек сообщений фиксированной длины из узла A в узел B вычисляется среднее (или медиана), стандартное отклонение, максимум и минимум. При этом выбор вычисляемых статистик не основывается на свойствах и природе описываемых распределений, и значит их набор возможно не является наиболее эффективным описанием нужных распределений.

Первая часть данной работы будет посвящена построению модели задержек сообщений при фиксированных контролируемых пользователем параметрах: длине сообщения, узле-отправителе и узле-получателе. Во второй части работы будет предложен метод диагностики вычислительного кластера на основании коммуникационных свойств задержек. В третьей части будет предложен метод кластеризации отдельных конкретных задержек для повышения эффективности их использования.

Все предложенные методы будут проиллюстрированы реальными задержками сообщений в суперкомпьютерных системах МГУ им. М. В. Ломоносова.

Глава 2

Анализ индивидуальных задержек

2.1 Модель вычислительного кластера

Опишем для начала общую модель вычислительной системы по [13].

Всё разнообразие архитектур вычислительных систем можно, на основании наличия параллелизма в потоках данных и в потоках команд, разбить на 4 класса:

- Системы с одиночным потоком команд и одиночным потоком данных (SISD, Single Instruction stream over a Single Data stream).

К этому классу относятся, прежде всего, классические последовательные машины. В таких машинах есть только один поток команд, команды обрабатываются последовательно и каждая команда описывает одну операцию с одним потоком данных.

- Системы с одиночным потоком команд и множественным потоком данных (SIMD, Single Instruction, Multiple Data).

В системах этого класса сохраняется один поток команд, включающий, в отличие от систем класса SISD, векторные команды. Это позволяет выполнять одну операцию сразу над многими данными — элементами вектора.

- Системы со множественным потоком команд и одиночным потоком данных (MISD, Multiple Instruction Single Data).

В системах этого класса подразумевается наличие в системе многих вычислительных устройств, обрабатывающих один общий поток данных. Но

исследователям не удалось предложить пример реальной вычислительной системы, удовлетворяющей этому описанию, поэтому данный класс принято считать пустым.

- Системы со множественным потоком команд и множественным потоком данных (MIMD, Multiple Instruction Multiple Data).

Этот класс предполагает, что в вычислительной системе есть несколько устройств обработки команд, объединённых в единый комплекс и работающих каждое со своим потоком команд и данных. Именно системы данного класса требуют реализации внутри себя протоколов передачи сообщений между устройствами, поэтому именно их мы будем рассматривать в данном исследовании.

В данном исследовании будут рассматриваться только вычислительные системы класса MIMD.

Все вычислительные системы класса MIMD в свою очередь по организации памяти делятся на два подкласса: вычислительные системы с общей памятью и вычислительные системы с распределённой памятью.

В системах из первого подкласса все процессоры работают в едином адресном пространстве памяти независимо от её физического расположения. В системах из второго подкласса память (в смысле адресного пространства) некоторым образом распределена между процессорами системы.

Таким образом, все процессоры вычислительной системы с распределённой памятью можно разбить на группы процессоров, работающих в одном адресном пространстве. Такие группы называются вычислительными узлами. Для обмена информацией узлы одной вычислительной системы объединяются друг с другом некоторой коммуникационной средой. Заметим, что процессоры имеют доступ к памяти только своего узла, и получение информации с других узлов возможно только через коммуникационную среду данной системы.

Например, установленная на ВМК МГУ им. М.В. Ломоносова вычислительная система Regatta является системой с общей памятью, а системы Blue Gene/P и Ломоносов — системами с распределённой памятью.

Если рассматривать узлы как вершины, а соединения в рамках коммуникационной среды между ними как рёбра, мы получим коммуникационную сеть вычислительной системы.

Определение 1. *Вычислительный узел — группа процессоров, работающих в едином адресном пространстве. Обычно представляет собой отдельную вычислительную машину под управлением своей операционной системы.*

Определение 2. *Коммуникационная сеть — множество вычислительных узлов и коммуникационных связей между ними.*

Определение 3. *Вычислительный кластер — объединённый коммуникационной сетью набор узлов, выполняющий вычисления и представляемый пользователю как единая система.*

На каждом узле установлена операционная система, управляющая работой на нём процессов. Процесс в данном случае — это выполняющаяся программа вместе со всеми её элементами (адресное пространство, регистры, стек и так далее). При распараллеливании задачи для систем с распределённой памятью она разбивается на несколько выполняющихся в виде процессов на отдельных узлах подзадач. Естественно, в большинстве случаев необходимо организовывать передачу данных между процессами.

Существует много программных интерфейсов для коммуникации процессов, запущенных на узлах системы. Самая распространённая из таких технологий — это MPI [5]. MPI рассматривает все коммуникации между процессами на узлах как сообщения определённой структуры: либо стандартной (для стандартных типов), либо определённой разработчиком.

В нашем исследовании мы не будем подробно останавливаться на исследовании влияния структуры сообщений на их передачу через коммуникационную сеть и будем варьировать только размер сообщений, узел-отправитель и узел-получатель. Таким образом, для нас сообщение — это набор байтов определённой длины с определённым узлом-отправителем и узлом-получателем.

Определение 4. *Сообщение в коммуникационной сети вычислительного кластера — это набор байтов определённой длины, передающийся от определённого узла-*

отправителя к определённому узлу-получателю через коммуникационную сеть вычислительного кластера.

Определение 5. *Контролируемыми параметрами сообщения мы будем называть длину сообщения, идентификатор узла-получателя и идентификатор узла-отправителя.*

Напомним, что в данном исследовании мы абстрагируемся от вопроса физической природы задержек. Сообщение для нас — некоторый объект, описываемый своими контролируемыми параметрами. И наша задача — исследовать свойства данного объекта на выбранном нами уровне абстракции.

2.2 Модель задержек

Рассмотрим многомерный массив данных, измерениями которого являются уже упомянутые нами выше контролируемые параметры сообщения: процесс-отправитель, процесс-получатель и длина. Пусть мы работаем с фиксированной вычислительной системой. Определим понятие задержки сообщения.

Определение 6. *Задержка сообщения в коммуникационной сети — это длина интервала времени с момента полной отправки сообщения узлом-отправителем до момента его получения узлом-получателем.*

В конкретных вычислительных системах с помощью специальных программных продуктов можно измерять определённые нами задержки сообщений при фиксированных контролируемых параметрах. Таким образом, в каждой ячейке многомерного массива мы получаем некоторый набор задержек. Из определения очевидно, что задержки являются неотрицательными действительными числами. В данной главе нашей задачей является построение наиболее оптимального описания набора задержек независимо в каждой ячейке массива данных, то есть при фиксированных контролируемых параметрах сообщения.

Выше уже было сказано, что задержки зависят не только от рассматриваемых нами параметров сообщения, но и от многих других неизвестных нам параметров, описывающих состояние сети. Этот факт позволяет нам смоделировать задержку в

каждой отдельной ячейке массива данных случайно величиной с некоторым известным нам распределением.

Определение 7. *Модель задержек сообщений в коммуникационной сети — это случайная величина, описывающая задержки сообщений с фиксированными контролируемыми параметрами.*

В некоторых работах уже предпринимались попытки создание и описание моделей задержек. Например, в [9] производился анализ задержек пакетов IP в сети Интернет и получилось, что большинство исследованных автором распределений задержек близки либо к трёхпараметрическому гамма-распределению, либо к трёхпараметрическому логнормальному распределению в зависимости от получателя и отправителя. Также в [3] и в [10] было показано, что задержки хорошо описываются трёхпараметрическим логнормальным распределением. Отметим, что условия проведения исследований во всех рассматриваемых работах (исследования проводились на задержках сообщений в сети Интернет) отличаются от таковых в нашем случае.

2.3 Описание данных

Данные для проведения исследования были собраны на вычислительных системах, установленных в МГУ им. М. В. Ломоносова.

Как уже было подробно описано выше, структура графа коммуникационной сети вычислительного кластера различна у разных вычислительных систем. Кратко опишем вычислительные системы, с которыми мы имели дело в рамках исследования:

- вычислительный комплекс Regatta: 1 вычислительный узел, 16 процессоров
- вычислительный комплекс IBM Blue Gene/P: 2048 вычислительных узлов, 8192 ядер
- суперкомпьютер «Ломоносов»: 6654 вычислительных узла, более 94000 процессорных ядер

На описанных выше вычислительных системах были собраны следующие данные:

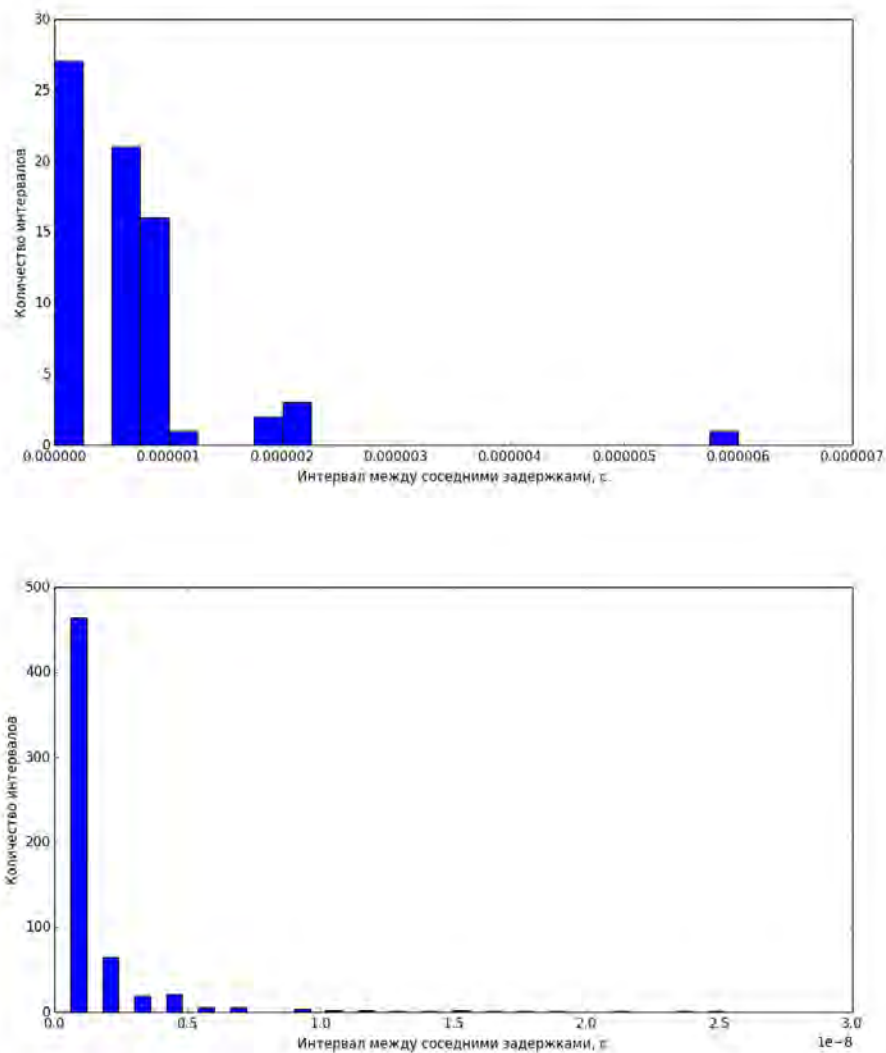


Рис. 2.1: Нерегулярность интервалов между уникальными значениями задержек: сверху задержки с СК «Ломоносов», снизу с вычислительного комплекса Blue Gene/P

- На вычислительной системе Regatta и на суперкомпьютере «Ломоносов» было собрано по 100000 задержек для сети из 8 процессов для сообщений длинами от 5000 до 100000 байтов с шагом в 5000 байтов.
- На суперкомпьютере IBM Blue Gene/P было собрано
 - по 10000 задержек для сети из 8 процессов и для сообщений длинами от 0 байтов до 500 байтов с шагом в 4 байтов.
 - по 10000 задержек для сети из 8 процессов и для сообщений длинами от 500 байтов до 10000 байтов с шагом в 500 байтов.

- по 10000 задержек для сети из 8 процессов и для сообщений длинами от 10000 байтов до 100000 байтов с шагом в 5000 байтов.
- по 10000 задержек для сетей из 16, 32, 64 процессов и для сообщений с длинами от 0 до 1000 с шагом 50 байтов.

Также стоит отметить, что данные для предварительного тестирования были собраны с персонального компьютера.

Изучение собранных данных позволяет заметить некоторые особенности изучаемых нами задержек сообщений.

Это, во-первых, «мульти-modalность» распределений задержек: на подавляющем большинстве гистограмм можно визуально выделить несколько характерных «горбов» (обычно от одного до трёх, примеры можно видеть на рис. 2.2). В дальнейшем это свойство повлияет на выбор семейства распределений для моделирования задержек.

Во-вторых, это «дискретность» задержек: число уникальных значений задержек в выборке много меньше длины выборки. Задержки делятся на довольно большие группы равных между собой, как если бы их «округляли» после измерения. Например:

- Среди задержек с суперкомпьютера «Ломоносов» сообщений длины 50000 из 0 узла в 4 узел всего 73 уникальных значения из 100000 задержек.
- Среди задержек с суперкомпьютера Blue Gene/P сообщений длины 500 из 3 узла в 1 узел 608 уникальных значений из 10000 задержек.

Также можно отметить нерегулярность интервалов между уникальными значениями задержек. Примеры гистограмм таких интервалов можно увидеть на рис. 2.1.

2.4 Технология сбора данных

Сбор данных производился с помощью программной системы Network Test из пакета Parus [12]. Данная система представляет собой инструмент для тестирования и анализа коммуникационной сети вычислительных систем с помощью программного интерфейса MPI [5].

Система Network Test позволяет анализировать коммуникационную сеть вычислительного кластера в нескольких режимах: one-to-one, режим рассылки, режимы get и put и так далее. Для получения данных использовался только режим one-to-one. В этом режиме один процесс посылает сообщение одному другому процессу, а другие процессы в это время бездействуют.

В рамках работы также была предложена и внедрена модульная архитектура для системы Network Test.

2.5 Обзор моделей распределений задержек

Как уже было сказано выше, задержки в общем случае имеют мультимодальное распределение с несколькими «горбами» (чаще всего не больше 3). Также описанный выше обзор литературы показал, что задержки сообщений хорошо описываются трёхпараметрическим логнормальным распределением. Заметим, что визуальный анализ наших данных показал мультимодальность распределений задержек. Логично было бы попробовать описать полученные данные смесью нескольких трёхпараметрических логнормальных распределений.

Трёхпараметрическое логнормальное распределение задаётся параметрами: $\sigma > 0$ (параметр формы), μ (параметр масштаба) и $\xi \geq 0$ (параметр сдвига). Для $x \leq \xi$ его плотность равна 0, а для $x > \xi$

$$f(x; \xi, \mu, \sigma) = \frac{1}{(x - \xi)\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x-\xi)-\mu)^2}{2\sigma^2}} \quad (2.1)$$

Если случайная величина x имеет трёхпараметрическое логнормальное распределение с параметрами σ , μ и ξ , то $\ln(x - \xi)$ имеет нормальное распределение с параметрами μ и σ .

В [6] было показано, что оценки максимального правдоподобия для параметров трёхпараметрического логнормального распределения расходятся: $\hat{\xi} = x_{(1)}$, $\hat{\sigma} = +\infty$, $\hat{\mu} = +\infty$.

2.6 Методы оценки параметров модели

Целью данного раздела является обзор методов оценки параметров выбранных распределений по описанным выше реальным данным. Заметим, что для подбора подходящего метода надо учесть и особенности данных, и свойства распределений, параметры которых мы пытаемся оценить. Выше мы решили, что эти распределения представляют собой смесь нескольких трёхпараметрических логнормальных распределений, а наиболее распространённым методом оценки параметров смесей является EM-алгоритм. При применении EM-алгоритма для решения нашей задачи мы столкнулись с некоторыми трудностями, связанными в том числе с упомянутыми нами проблемами в оценках максимального правдоподобия параметров трёхпараметрического логнормального распределения. Поэтому нами был предложен альтернативный метод.

2.6.1 Постановка задачи

Итак, было принято решение описывать задержки смесью нескольких (от 1 до 3) трёхпараметрических логнормальных распределений. Теперь необходимо выбрать метод разделения соответствующей смеси, то есть оценки параметров соответствующей модели.

Сформулируем задачу. Нам дана сгенерированная из неизвестного нам распределения выборка $x_i, i = 1, 2, \dots, N$. Мы хотим приблизить неизвестное распределение смесью из нескольких трёхпараметрических логнормальных распределений. Требуется предложить метод оценки параметров данной смеси.

2.6.2 EM-алгоритм

Часто рассматриваемая нами в данном разделе задача решается с помощью EM-алгоритма.

Напомним, что идея применения EM-алгоритма к задачам разделения смеси распределений состоит во введении в модель скрытых переменных g_{ij} , упрощающих вычисления. Напомним также, что EM-алгоритм решает задачу максимизации прав-

доподобия, проблемы с которой применительно к нашим распределениям были упомянуты выше.

Одна итерация EM-алгоритма состоит из двух шагов: на первом (expectation-шаге) производится оценка вектора скрытых переменных, а на втором (maximization-шаге) максимизируется математическое ожидание по скрытым переменным логарифма правдоподобия модели.

Пусть q_{nk} — введённые скрытые переменные, $f(x; \theta)$ — плотность логнормального распределения с параметрами $\theta = (\xi, \mu, \sigma)$, w_k — вес k -й компоненты смеси, $k = 1, \dots, K$ и x_i , $i = 1, \dots, N$ — выборка. Тогда применительно к нашей задаче один шаг EM-алгоритма выглядят следующим образом:

1. E-шаг, оценка скрытых переменных:

$$q_{nk} = \frac{w_k f(x_n; \theta_k)}{\sum_{j=1}^K w_j f(x_n; \theta_j)} \quad n = 1, \dots, N, k = 1, \dots, K \quad (2.2)$$

2. M-шаг, переоценка параметров:

$$w_k = \frac{1}{N} \sum_{i=1}^n q_{ik}, \quad k = 1, \dots, K \quad (2.3)$$

$$\theta_k = \arg \max_{\theta} \sum_{i=1}^n q_{ik} \ln f(x_i; \theta_k), \quad k = 1, \dots, K \quad (2.4)$$

Заметим, что на M-шаге в 2.4 решается задача максимизации взвешенного правдоподобия, а как уже было сказано выше, оценки максимального правдоподобия для рассматриваемых распределений расходятся.

2.6.3 Метод минимизации расстояния

Как будет показано ниже, в экспериментальном разделе, в нашей задаче использовать EM-алгоритм для оценки параметров смеси нецелесообразно, поэтому был предложен другой метод.

Пусть $F_n(x)$ — эмпирическая функция распределения, $F(x, \theta)$ — функция распределения модели, а $d[\cdot, \cdot]$ — функционал «расстояния» между двумя распределениями. Тогда оценка минимума расстояния на параметры $\theta \in \Theta$, это такое $\hat{\theta}$, что

$$d[F(x, \hat{\theta}), F_n(x)] = \min_{\theta \in \Theta} d[F(x, \theta), F_n(x)] \quad (2.5)$$

В качестве расстояния чаще всего используются статистики критериев согласия, а именно:

- статистика критерия Колмогорова-Смирнова

$$D_n(x) = \frac{1}{n} \max_i |F_n(x_i) - F(x_i, \theta)| = \frac{1}{n} \max_i \left| \frac{1}{n} - F(x_i, \theta) \right| \quad (2.6)$$

- статистика критерия Крамера-фон Мизеса

$$T_n(x) = \frac{1}{12n} + \sum_{i=1}^n \left[\frac{2i-1}{2n} - F(x, \theta) \right]^2 \quad (2.7)$$

- статистика критерия Андерсона-Дарлинга

$$A_n(x) = -n - \sum_{i=1}^n \frac{2i-1}{n} (\ln F(x_i, \theta) + \ln(1 - F(x_i, \theta))) \quad (2.8)$$

и их модификации.

В [4] показано, что в большинстве случаев (авторами тестировалось в том числе выборки из логнормального распределения) наилучшую точность и сходимость даёт использование взвешенного расстояния Крамера - фон Мизеса

$$T_n^*(x) = \frac{1}{12n} + \sum_{i=1}^n \frac{1}{\frac{2i-1}{2n} \cdot (1 - \frac{2i-1}{2n})} \left[\frac{2i-1}{2n} - F(x, \theta) \right]^2 \quad (2.9)$$

$$= \frac{1}{12n} + \sum_{i=1}^n \frac{2n}{(2i-1)(2n-2i-1)} \left[\frac{2i-1}{2n} - F(x, \theta) \right]^2 \quad (2.10)$$

Таким образом, для оценки параметров модели необходимо с помощью одного из методов оптимизации решить задачу

$$\hat{\theta} = \arg \min_{\theta \in \Theta} d[F(x_{(i)}, \theta), F_n(x_{(i)})] \quad (2.11)$$

$$= \arg \min_{\theta \in \Theta} \left(\frac{1}{12n} + \sum_{i=1}^n \frac{2n}{(2i-1)(2n-2i-1)} \left[\frac{2i-1}{2n} - F(x_{(i)}, \theta) \right]^2 \right) = \quad (2.12)$$

$$= \arg \min_{\theta \in \Theta} \left(\sum_{i=1}^n \frac{2n}{(2i-1)(2n-2i-1)} \left[\frac{2i-1}{2n} - F(x_{(i)}, \theta) \right]^2 \right) \quad (2.13)$$

2.7 Тестирование методов оценки параметров модели

2.7.1 Предобработка данных

Итак, нам необходимо оценить параметры смеси трёхпараметрических логнормальных распределений по собранным данным. При этом необходимо учесть уже описанные выше особенности данных, а именно их «дискретность». Предварительное тестирование показало, что применение EM-алгоритма или метод минимизации расстояния между модельным и эмпирическим распределениями напрямую к данным, без какой-либо их предварительной обработки, не приводит к желаемому результату: компоненты смеси в этом случае сходятся не к визуально различимым на графиках «горбам», а к отдельным самым крупным группам равных задержек. С подобным эффектом приходилось, конечно, сталкиваться и другим исследователям: например, он описан в [11].

Для устранения подобных нежелательных эффектов было решено провести предобработку данных. В ходе предварительного тестирования было решено добавить в данные перед запуском на них соответствующих методов гауссовский шум. Для выбора уровня шума (дисперсии соответствующего нормального распределения) на каждом наборе данных проводилось тестирование методов на одной из выборок при разном уровне шума. Выбирался уровень шума, при котором достигается наиболее оптимальное значение расстояния.

2.7.2 Результаты тестирования

Тестирование обоих методов (EM-алгоритма и метода минимизации расстояния) производилось на описанных выше реальных данных. Они были реализованы на языке Python с использованием библиотек NumPy, SciPy [8] и NLOpt [7]. В качестве метода оптимизации использовалась реализация метода PRAXIS (подробное описание можно найти например в [2]) из библиотеки NLOpt. В методах фиксировалось общее для всех методов значение допуска на относительное изменение функции $\frac{\delta f}{f_{\text{cur}}}$. Оно выбиралось из соображений разумности времени работы алгоритмов на тех данных,

на которой проводились эксперименты. Начальное приближение для обоих алгоритмов выбиралось с помощью EM-алгоритма для смеси нормальных распределений.

Примеры работы методов можно видеть на рис. 2.3, рис. 2.4 и рис. 2.5.

Как видно, эксперименты показали, что метод минимизации расстояния работает быстрее и лучше EM-алгоритма.

2.8 Выводы

В рамках данной части исследования был проведён сбор данных, первоначальный (визуальный) анализ индивидуальных (при фиксированных характеристиках сообщений) задержек. На основе существующих исследований была предложена модель индивидуальных задержек и метод оценки параметров данного метода. Также было проведено тестирование предложенного метода на модельных данных.

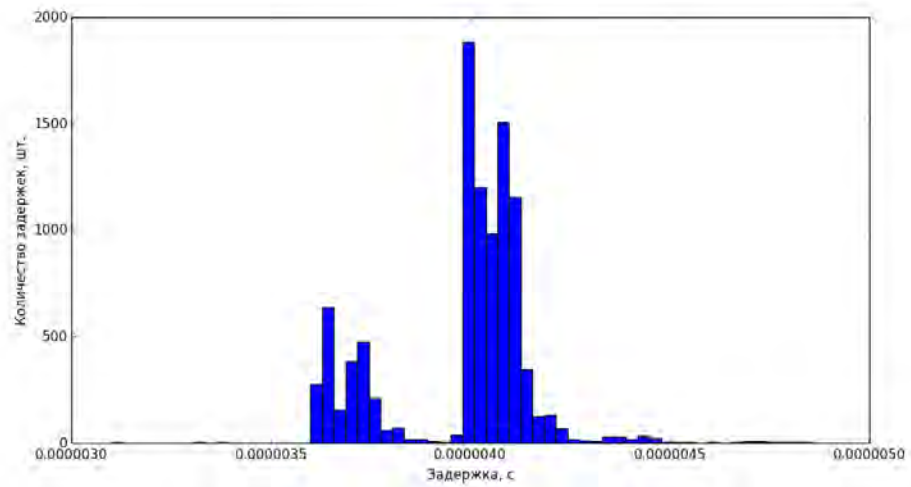
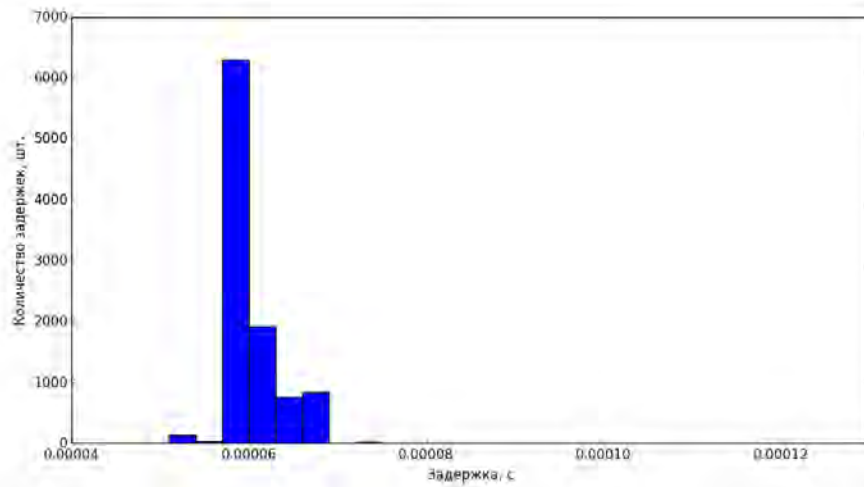
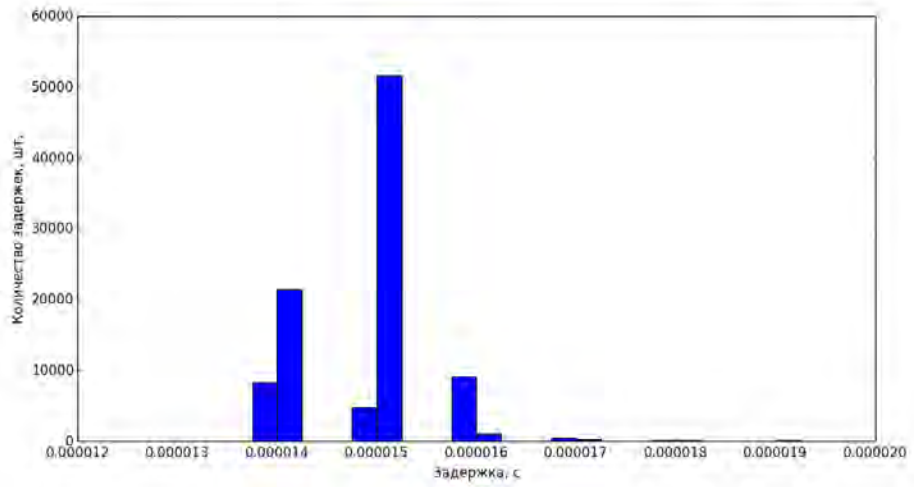
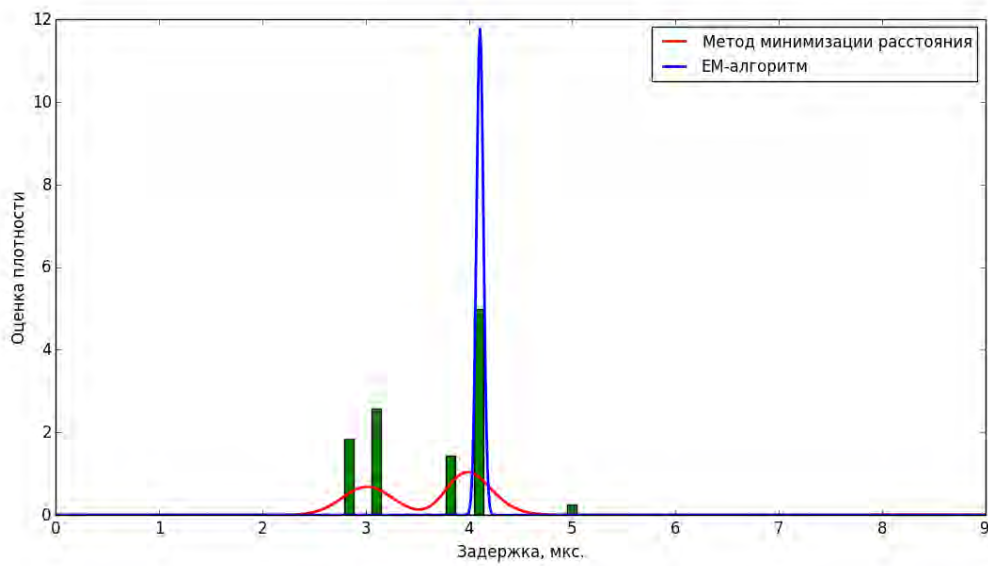
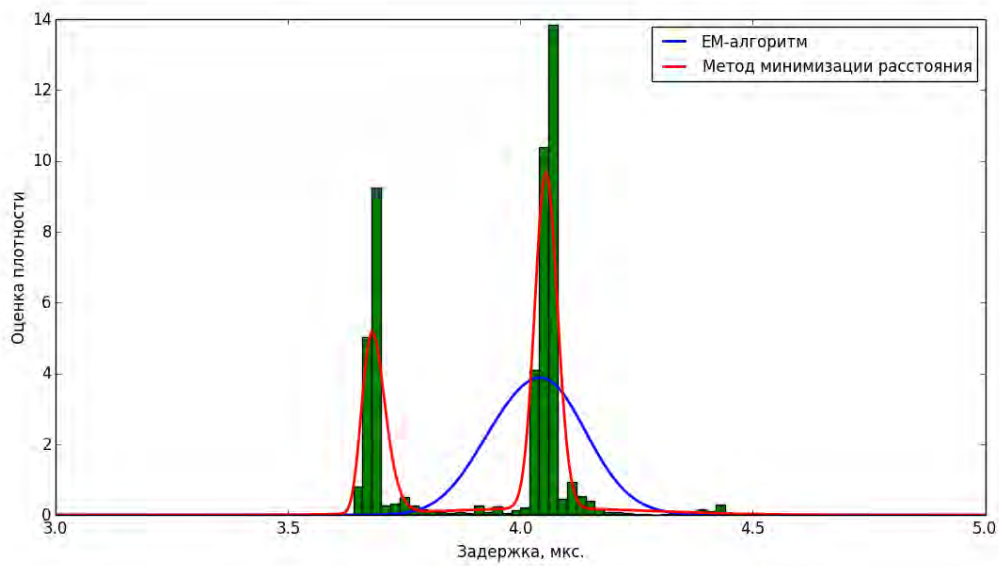


Рис. 2.2: Мульти-modalность распределений задержек: сверху вниз соответственно данные с Ломоносова, Regatta и Blue Gene/P



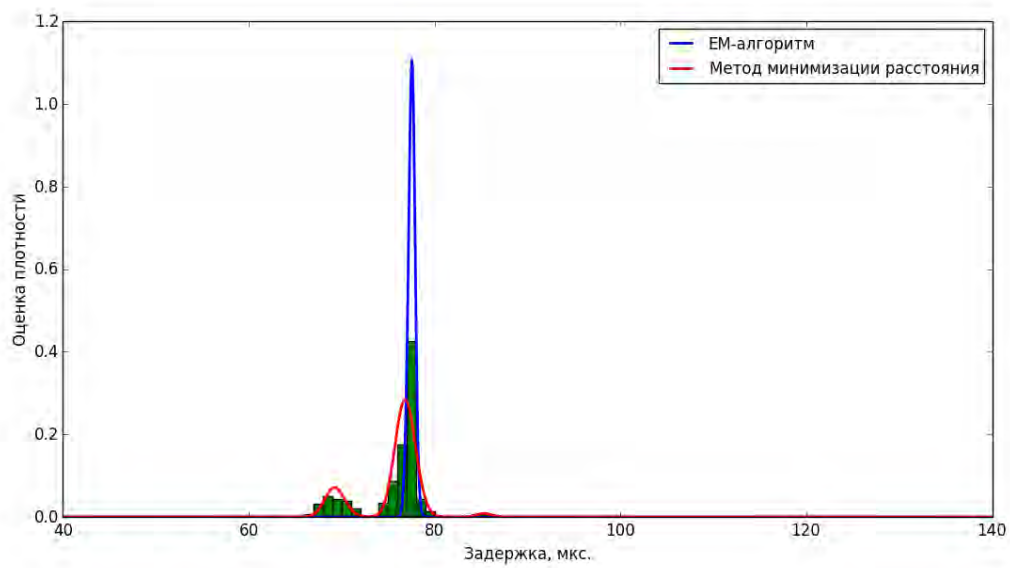
	EM-алгоритм	Метод минимизации расстояния
Время работы	75.86 с.	31.97 с.
$\frac{1}{n} \ln L(x)$	-1.1190	-0.45025
$\frac{1}{n} d[F(x), F_n]$	0.5991	0.1482

Рис. 2.3: Пример работы алгоритмов на задержках с СК «Ломоносов»: в таблице приведено время работы, усреднённый логарифм правдоподобия и усреднённое расстояние между распределениями.



	EM-алгоритм	Метод минимизации расстояния
Время работы	111.1760 с.	4.54 с.
$\frac{1}{n} \ln L(x)$	0.9773	1.5194
$\frac{1}{n} d[F(x), F_n]$	0.1235	0.0330

Рис. 2.4: Пример работы алгоритмов на задержках с системы Blue Gene/P: в таблице приведено время работы, усреднённый логарифм правдоподобия и усреднённое расстояние между распределениями.



	EM-алгоритм	Метод минимизации расстояния
Время работы	31.065 с.	3.287 с.
$\frac{1}{n} \ln L(x)$	-2.2358	-2.2063
$\frac{1}{n} d[F(x), F_n]$	0.1401	0.1340

Рис. 2.5: Пример работы алгоритмов на задержках с системы Regatta: в таблице приведено время работы, усреднённый логарифм правдоподобия и усреднённое расстояние между распределениями.

Глава 3

Анализ задержек на маршрутах

В данной главе будут выведены и изучены на реальных данных некоторые связанные с природой задержек как некоторых порций данных, «перемещающихся» по коммуникационной сети, свойства задержек, выполнения которых было бы естественно потребовать. Информация об их выполнении или не выполнении на может помочь выявлению и диагностики неисправностей вычислительной системы.

3.1 Свойства монотонности, неравенства треугольника и неделимости

Пусть $t_{s,a \rightarrow b}$ — задержка сообщений длины s от узла a до узла b . Основываясь на природе задержек, логично потребовать выполнения следующих трёх неравенств.

$$1. t_{s_1,a \rightarrow b} > t_{s_2,a \rightarrow b}, s_1 > s_2 \quad (\text{свойство монотонности}) \quad (3.1)$$

$$2. t_{s,a \rightarrow b} \leq t_{s,a \rightarrow c} + t_{s,c \rightarrow b} \quad (\text{неравенство треугольника}) \quad (3.2)$$

$$3. t_{s_1+s_2,a \rightarrow b} \leq t_{s_1,a \rightarrow b} + t_{s_2,a \rightarrow b} \quad (\text{свойство неделимости}) \quad (3.3)$$

Утверждение 1. *Все три рассматриваемые свойства (монотонности, неделимости и неравенство треугольника) независимы.*

Доказательство. Докажем сначала независимость свойства монотонности и неравенства треугольника. Рассмотрим модельную вычислительную систему из 3 узлов,

и задержки для двух длин сообщений s_1 и s_2 . Пусть $s_1 > s_2$. Пусть задержки имеют следующие значения:

$$t_{s_1,0 \rightarrow 1} = 5, \quad t_{s_2,0 \rightarrow 1} = 3 \quad (3.4)$$

$$t_{s_1,0 \rightarrow 2} = 2, \quad t_{s_2,0 \rightarrow 2} = 1 \quad (3.5)$$

$$t_{s_1,2 \rightarrow 1} = 2, \quad t_{s_2,2 \rightarrow 1} = 1 \quad (3.6)$$

Заметим, что в этом случае выполняется и свойство монотонности, и неравенство треугольника. Рассмотрим теперь другие значения задержек:

$$t_{s_1,0 \rightarrow 1} = 5, \quad t_{s_2,0 \rightarrow 1} = 3 \quad (3.7)$$

$$t_{s_1,0 \rightarrow 2} = 3, \quad t_{s_2,0 \rightarrow 2} = 2 \quad (3.8)$$

$$t_{s_1,2 \rightarrow 1} = 3, \quad t_{s_2,2 \rightarrow 1} = 2 \quad (3.9)$$

Заметим, что в данном случае свойство монотонности выполняется, но неравенство треугольника нарушено.

$$t_{s_1,0 \rightarrow 1} = 3, \quad t_{s_2,0 \rightarrow 1} = 5 \quad (3.10)$$

$$t_{s_1,0 \rightarrow 2} = 2, \quad t_{s_2,0 \rightarrow 2} = 3 \quad (3.11)$$

$$t_{s_1,2 \rightarrow 1} = 2, \quad t_{s_2,2 \rightarrow 1} = 3 \quad (3.12)$$

В данном примере нарушены оба свойства. И, наконец, последний случай.

$$t_{s_1,0 \rightarrow 1} = 3, \quad t_{s_2,0 \rightarrow 1} = 5 \quad (3.13)$$

$$t_{s_1,0 \rightarrow 2} = 1, \quad t_{s_2,0 \rightarrow 1} = 2 \quad (3.14)$$

$$t_{s_1,2 \rightarrow 1} = 1, \quad t_{s_2,0 \rightarrow 1} = 2 \quad (3.15)$$

Здесь, очевидно, выполняется неравенство треугольника и нарушено свойство монотонности.

Итак, неравенство треугольника и свойство монотонности независимы.

Докажем теперь независимость свойства монотонности и свойства неделимости. Рассмотрим модельную систему с 2 узлами. Будем без потери общности рассматривать задержки длин 1 и 2. Как и в прошлой части доказательства, построим примеры к 4 возможным ситуациям:

1. Свойство монотонности выполняется, свойство неделимости не выполняется.

$$t_{1,0 \rightarrow 1} = 0, \quad t_{2,0 \rightarrow 1} = 1 \quad (3.16)$$

2. Оба свойства выполняются.

$$t_{1,0 \rightarrow 1} = 2, \quad t_{2,0 \rightarrow 1} = 3 \quad (3.17)$$

3. Оба свойства не выполняются (рассмотрим дополнительную длину сообщения 3)

$$t_{1,0 \rightarrow 1} = 2, \quad t_{2,0 \rightarrow 1} = 1, \quad t_{3,0 \rightarrow 1} = 6 \quad (3.18)$$

4. Свойство монотонности не выполняется, свойство неделимости выполняется.

$$t_{1,0 \rightarrow 1} = 2, \quad t_{2,0 \rightarrow 1} = 1 \quad (3.19)$$

Очевидно, добавив ещё одну вершину в модель из пунктов 1 и 2 и соответствующим образом расставив задержки, можно варьировать справедливость неравенства треугольника. Рассмотрим, например, пункт первый. Пусть дополнительно к уже указанным существуют задержки $t_{2,0 \rightarrow 2} = t_{2,2 \rightarrow 1} = 0.1$. В этом случае неравенство треугольника не выполняется. Если же $t_{2,0 \rightarrow 2} = t_{2,2 \rightarrow 1} = 0.6$, неравенство выполняется. Аналогично с пунктом 2.

Итак, мы доказали независимость всех трёх свойств. □

3.2 Проверка свойств на полученных данных

Как уже было сказано выше, мы рассматриваем задержки как случайные величины с некоторым неизвестным нам распределением. Таким образом, необходимо понять, как проверять выполнения неравенств для распределений.

Будем считать, что неравенства $x > y$ или $x \geq y$ справедливы, если соответственно $P(x > y) > \frac{1}{2}$ или $P(x > y) \geq \frac{1}{2}$. Аналогично для противоположных неравенств $P(x > y) < \frac{1}{2}$ и $P(x > y) \leq \frac{1}{2}$.

3.2.1 Проверка свойств на выборках

Сначала проверим данные свойства в предположении, что мы имеем дело с исходными выборками задержек. Для проверки соответствующих неравенств по выборкам мы воспользуемся ранговым критерием Уилкоксона-Манна-Уитни. Подробное его описание можно найти в [1].

В случае использования выбранного теста для проверки неравенства треугольника и свойства неделимости необходимо сгенерировать выборку для суммы задержек $X + Y$. Для генерации выборки для суммы длины N воспользуемся следующим методом:

- Сгенерируем выборку $(a_i, b_i)_{i=1}^N$ из равномерного распределения на множестве $\{(a, b) \mid a \in \{1, \dots, l_x\}, b \in \{1, \dots, l_y\}\}$, где l_x и l_y — длины выборок X и Y .
- По выборке $(a_i, b_i)_{i=1}^N$ составим выборку $(x_{a_i} + y_{b_i})_{i=1}^N$

Заметим, что в случае невыполнения нулевой гипотезы статистика $\frac{W - m_W}{\sigma_W}$ (которая при выполнении гипотезы $P(x < y) = \frac{1}{2}$ распределена нормально $\mathcal{N}(0, 1)$) отклоняется в экстремальные значения в ту или иную сторону в зависимости от отклонения $P(x < y)$ от $\frac{1}{2}$ в ту или иную сторону.

Описанным методом были протестированы данные с Blue Gene/P, СК «Ломоносов» и системы Regatta. Результаты тестирования можно видеть в таблицах 3.1, 3.2 и 3.3.

Вид теста	Нарушений	Мин. статистика	Макс. статистика	Всего тестов
Монотонность	297	-122.298	122.471	7728
Неравенство треугольника	0	121.882	122.471	1344
Неделимость	1146	-122.471	122.471	3696

Таблица 3.1: Результаты тестирования системы Regatta (уровень значимости 0.05).

Вид теста	Нарушений	Мин. статистика	Макс. статистика	Всего тестов
Монотонность	53746	-122.471	122.471	513408
Неравенство треугольника	0	17.316	122.471	10752
Неделимость	106862	-122.471	122.471	255360

Таблица 3.2: Результаты тестирования СК «Ломоносов» (уровень значимости 0.05).

Вид теста	Нарушений	Мин. статистика	Макс. статистика	Всего тестов
Монотонность	9044	-103.498	122.471	321328
Неравенство треугольника	0	122.444	122.471	8512
Неделимость	0	122.323	122.471	46564

Таблица 3.3: Результаты тестирования Blue Gene/P (уровень значимости 0.05).

3.2.2 Проверка свойств на моделях

Теперь будем проверять вышеперечисленные свойства с помощью построенных в первой части моделей. Учитывая независимость задержек, распишем необходимые нам неравенства.

$$P(x > y) = \int_{-\infty}^{+\infty} \int_{-\infty}^x p_x(\mu) p_y(\xi) d\xi d\mu = \quad (3.20)$$

$$= \left\{ p_x(\mu) = \sum_i w_i p_{x,i}(\mu), p_y(\xi) = \sum_j u_j p_{y,j}(\xi) \right\} = \quad (3.21)$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^x \left(\sum_i w_i p_{x,i}(\mu) \right) \left(\sum_j u_j p_{y,j}(\xi) \right) d\xi d\mu = \quad (3.22)$$

$$= \sum_i \sum_j w_i u_j \int_{-\infty}^{+\infty} p_{x,i}(\mu) \int_{-\infty}^x p_{y,j}(\xi) d\xi d\mu = \quad (3.23)$$

$$= \sum_i \sum_j w_i u_j \int_{-\infty}^{+\infty} p_{x,i}(\mu) F_{y,j}(\mu) d\mu \quad (3.24)$$

Заметим, что под последними интегралами стоят плотность трёхпараметрического логнормального распределения и его функция распределения $F(x; \xi, \mu, \sigma) = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left[\frac{\ln(x-\xi)-\mu}{\sqrt{2}\sigma} \right]$

Аналогичными выкладками получаем

$$P(x < y + z) = \left\{ \begin{array}{l} p_x(x) = \sum_i w_i p_{xi}(x), \\ p_y(y) = \sum_j u_j p_{yj}(y), \\ p_z(z) = \sum_k v_k p_{zk}(z) \end{array} \right\} = \quad (3.25)$$

$$= \sum_i \sum_j \sum_k w_i u_j v_k \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} p_{yj}(\tau) p_{zk}(\xi - \tau) d\tau \right] F_{x,i}(\xi) d\xi \quad (3.26)$$

Для проверки тестов на реальных задержках интегралы под суммами считались численно с помощью библиотек SciPy и GNU Scientific Library. Тестирование проводилось на всех собранных наборах данных. Результаты тестирования приведены на таблицах 3.4, 3.5 и 3.6.

Вид теста	Нарушений	Мин. вероятность	Макс. вероятность	Всего тестов
Монотонность	295	0.000	1.000	7728
Неравенство треугольника	0	0.996	1.000	1344
Неделимость	1146	0.000	1.000	3696

Таблица 3.4: Результаты тестирования системы Regatta.

Вид теста	Нарушений	Мин. вероятность	Макс. вероятность	Всего тестов
Монотонность	8212	0.031	1.000	165816
Неравенство треугольника	0	0.999	1.000	8512
Неделимость	0	0.992	1.000	46564

Таблица 3.5: Результаты тестирования Blue Gene/P.

Вид теста	Нарушений	Мин. вероятность	Макс. вероятность	Всего тестов
Монотонность	14065	0.000	1.000	155820
Неравенство треугольника	0	0.607	1.000	5936
Неделимость	27244	0.000	1.000	65688

Таблица 3.6: Результаты тестирования СК «Ломоносов».

3.2.3 Сравнение результатов проверок свойств на моделях и на данных

Теперь нам требуется узнать насколько совпадают результаты первого, использующего исходные выборки, метода проверки рассматриваемых свойств и второго, использующего только построенные модели. Для этого сравним результаты тестирования первым и вторым методами и посчитаем количество ошибок первого рода (первый метод показал выполнение свойства, второй — невыполнение) и второго рода (соответственно, наоборот). В таблицах 3.7, 3.8 и 3.9 приведены данные ошибки. Видно, что их очень мало. Таким образом, для тестирования коммуникационной сети вместо полных выборок можно использовать агрегированную информацию о задержках в форме параметров моделей.

Вид теста	Ошибок 1 рода	Ошибок 2 рода	Общих тестов
Монотонность	2	4	7728
Неравенство треугольника	0	0	1344
Неделимость	0	0	3696

Таблица 3.7: Сравнение результатов тестирования системы Regatta предложенными методами.

Вид теста	Ошибок 1 рода	Ошибок 2 рода	Общих тестов
Монотонность	553	1385	165816
Неравенство треугольника	0	0	8512
Неделимость	0	0	46564

Таблица 3.8: Сравнение результатов тестирования Blue Gene/P предложенными методами.

3.3 Выводы

В данной главе были предложены некоторые неравенства на задержках, проверка которых могла бы помочь в диагностике и выявлении неисправностей в коммуникационной сети вычислительного кластера. Выведенные неравенства были рассмотрены в терминах предлагаемого подхода к анализу задержек. Также были предложены и протестированы на реальных данных методы проверки данных неравенств и было показано, что переход от хранения полных выборок к хранению только параметров моделей не приводит к ошибкам в предложенном методе диагностики.

Вид теста	Ошибок 1 рода	Ошибок 2 рода	Общих тестов
Монотонность	34	72	155820
Неравенство треугольника	0	0	5936
Неделимость	243	221	65688

Таблица 3.9: Сравнение результатов тестирования СК «Ломоносов» предложенными методами.

Глава 4

Изучение зависимостей параметров моделей от характеристик сообщений

Ранее, в первой и второй частях данной работы, мы рассматривали задержки при фиксированных контролируемых параметрах сообщений. Возникает вопрос — а так ли необходимо для качественной оценки задержек хранить модели для всех возможных комбинаций контролируемых параметров? Возможно ли как-то агрегировать «похожие» описания задержек для некоторых групп контролируемых параметров?

4.1 Кластеризация пар узлов на основе данных о задержках

Рассмотрим наборы задержек при разных длинах сообщений, но при фиксированных узле-отправителе и узле-получателе. В этом случае каждой паре (узел-отправитель, узел-получатель) соответствует вектор задержек для разных длин сообщений. Нашей задачей в данной части исследования будет кластеризация пар узлов по поведению задержек при изменении (росте) длины сообщения.

Таким образом, каждая пара (a, b) описывается вектором $(t_{s_i, a \rightarrow b})_{i=1}^L$, где s_1, \dots, s_L — длины сообщений, для которых проводился сбор данных.

Заметим, что в поведении задержек при росте длины сообщения нас больше интересует рост их значений (см. например рис. 4.1). Напомним, что мы рассматриваем отдельную задержку как случайную величину, распределение которой мы научились

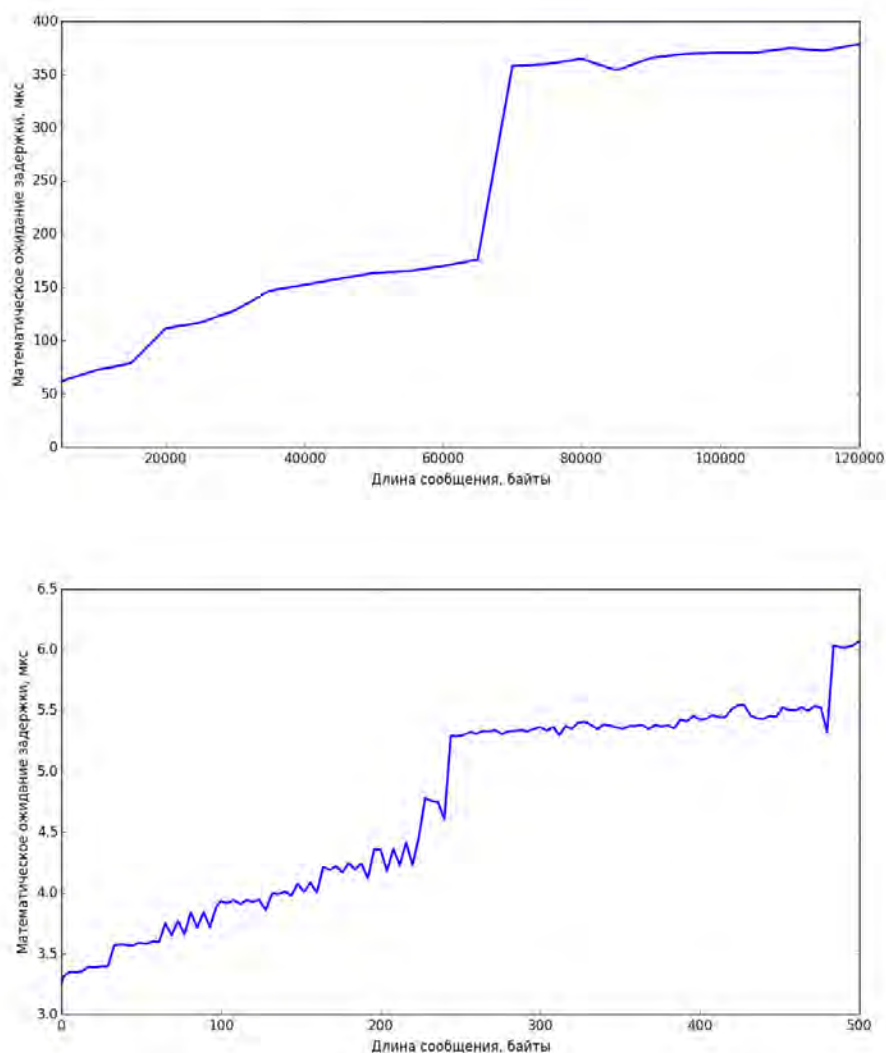


Рис. 4.1: Примеры роста математического ожидания распределений при увеличении длины сообщения: слева для отдельной задержки с системы Regatta, справа — с системы Blue Gene/P

приближать в предыдущих разделах. Таким образом, для описания изменения значения задержек как случайных величин для наших задач можно взять какой-либо коэффициент сдвига наших модельных распределений. В данном исследовании в качестве такого коэффициента сдвига используется математическое ожидание.

Итак, в качестве расстояния можно отдельными компонентами рассматриваемых векторов (которые, напомним, являются распределениями) возьмём модуль разности их математических ожиданий $\rho(t_{s,a \rightarrow b}, t_{s,c \rightarrow d}) = |\mathbf{E}t_{s,a \rightarrow b} - \mathbf{E}t_{s,c \rightarrow d}|$.

Отметим теперь, что нас интересуют различия (и сходства) в форме поведения задержек при изменении длины сообщения, схожесть аномалий этой формы, поэтому,

чтобы возможно было «ловить» эти аномалии, было решено в качестве расстояния взять максимум из попарных введённых нами расстояний между компонентами вектора.

Итак, суммируя всё вышесказанное, получаем:

$$\rho(a \rightarrow b, c \rightarrow d) = \max_{s \in \{s_1, \dots, s_L\}} |Et_{s, a \rightarrow b} - Et_{s, c \rightarrow d}| \quad (4.1)$$

Перейдём теперь к выбору подходящего для решения нашей задачи метода кластеризации.

4.1.1 Описание метода кластеризации

Заметим, что число обрабатываемых при кластеризации объектов квадратично зависит от исходного размера коммуникационной сети. Действительно, если n — число обменивающихся сообщениями процессов, то имеется $n(n-1)$ упорядоченных пар источник-приёмник (в случае совпадения источника и приёмника задержки равны нулю, поэтому такие пары мы убираем из рассмотрения), описываемых вектором из L задержек каждая. Таким образом, мы имеем $\frac{n(n-1)(n(n-1)-1)}{2}$ различных расстояний. Напомним, что современные кластеры могут состоять из тысяч и десятков тысяч узлов. Так что в выборе метода кластеризации не последнюю роль играет требования его эффективности.

Поэтому для решения данной задачи был выбран метод дивизивной кластеризации, позволяющий во-первых при некоторых эвристических модификациях избавиться от необходимости знания всех попарных расстояний, а во-вторых использовать на каждом следующем шаге уже посчитанные результаты предыдущего шага и считать необходимые задержки по необходимости.

Метод дивизивной кластеризации состоит в последовательном разбиении кластера на два кластера, основываясь на близости элементов к противоположным, наиболее удалённым, элементам исходного кластера.

Требования к «противоположности» зерён новых кластеров нестрогое, поэтому данную задачу можно решать приближённо. Например, хорошее приближение даёт следующая эвристика: рассмотрим произвольный элемент a из обрабатываемого кластера, найдём наиболее удалённый от него элемент $b = \arg \max_x \rho(x, a)$ из этого

же кластера, далее найдём наиболее удалённый от b элемент $c = \arg \max_x \rho(x, b)$ из кластера. Элементы b и c будем считать приближением противоположных элементов в обрабатываемом кластере.

Заметим, что при этом нам потребовалось посчитать всего $2n - 1$ расстояний, где n — размер кластера. Также заметим, что для проведения процедуры деления кластера на два необходимо посчитать дополнительно только $n - 1$ расстояние.

Таким образом, алгоритм дивизивной кластеризации в нашем случае состоит в следующем:

1. Изначально объединяем все объекты в один большой кластер.
2. Пока разница в диаметрах разделяемых кластеров не стабилизировалась (не стала существенно меньше типичного значения), повторяем:
3. (а) Выбираем кластер с наибольшим диаметром.
(б) Выбираем в нём противоположенные элементы b и c описанным выше способом.
(с) Делим рассматриваемый кластер по b и c , сохраняем диаметры ($\max_{x \in M_1} \rho(b, x)$ и $\max_{x \in M_2} \rho(c, x)$, где M_1 и M_2 — новые кластеры вокруг b и c соответственно). Заметим, что вычисление диаметров было выполнено на предыдущем шаге.

Также в листинге 1 приведён псевдокод алгоритма.

4.1.2 Тестирование метода

Метод был реализован на языке Python с использованием библиотек NumPy и SciPy.

Тестирование метода проводилось на всех собранных массивах данных. На графиках с рис. 4.2 и рис. 4.3 видно, что для данных с систем Regatta и Blue Gene/P и, пусть и с натяжкой, Ломоносова возможно выбрать оптимальное количество кластеров (то есть в некоторый момент происходит резкое уменьшение текущих радиусов кластеров и их стабилизация с медленным уменьшением), причём оптимальное (на котором происходит максимальное изменение радиуса кластеров и далее, при

Вход: $t_{s,a \rightarrow b}$, $a, b \in \{1, \dots, N\}$, $s \in \{s_1, s_2, \dots, s_L\}$ — модели задержек

Выход: Разбиение $\{1, \dots, N\}^2$ на кластеры C_1, \dots, C_K

$C_1 = \{1, \dots, N\}^2$;

до тех пор, пока радиусы r_i разбиваемых кластеров не стабилизируются

выполнять

$i = \arg \max_i r_i$, $i = 1, \dots, K$;

v — произвольный элемент C_i ;

$w = \arg \max_x \rho(x, v)$, $x \in C_i$;

$u = \arg \max_x \rho(x, w)$, $x \in C_i$;

$K = K + 1$;

$C_K = \{x \in C_i \mid \rho(x, u) < \rho(x, w)\}$, $r_K = \max_{a,b} \rho(a, b)$, $a, b \in C_K$;

$C_i = C_i \setminus C_K$, $r_K = \max_{a,b} \rho(a, b)$, $a, b \in C_K$;

конец цикла

Алгоритм 1: Дивизивная кластеризация пар узлов источник-приёмник

большем числе кластеров, стабилизация этого изменения) число кластеров для них невелико — для Regatta это 5 кластеров, для Blue Gene/P и Ломоносова 4. В таблице 4.1 приведены размеры получившихся кластеров, на рис. 4.4 приведены графики поведения математического ожидания типичных представителей кластеров для трёх систем.

Набор данных	Число кластеров	Размеры кластеров
Regatta	5	23, 13, 9, 3, 8
Blue Gene/P, 64 процесса	4	1584, 1356, 880, 212
Ломоносов	4	25, 22, 5, 4

Таблица 4.1: Результаты кластеризации на данных с Regatta, Blue Gene/P и Ломоносова.

4.2 Выводы

В данной части исследования был предложен метод агрегирования данных о моделях задержек с помощью кластеризации пар источник-приёмник, основанной на учёте различия поведения задержек при росте длин сообщений. Метод был реализо-

ван и протестирован на реальных данных, собранных с суперкомпьютерных кластеров МГУ им. М. В. Ломоносова.

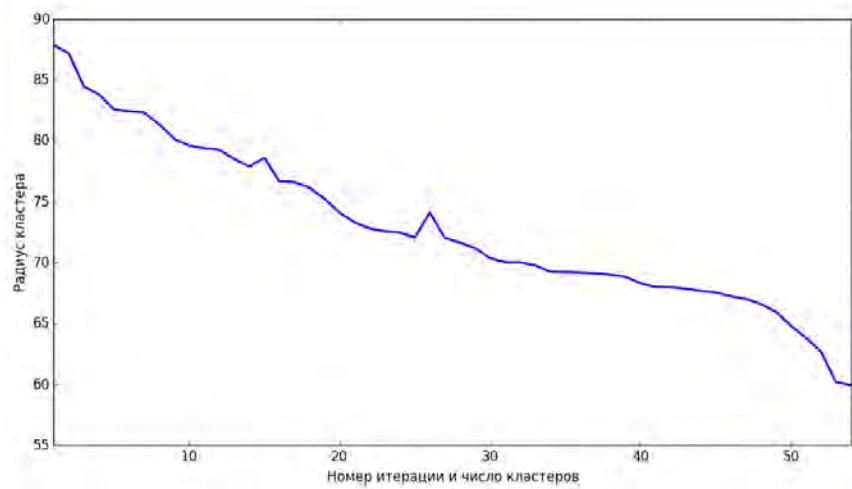
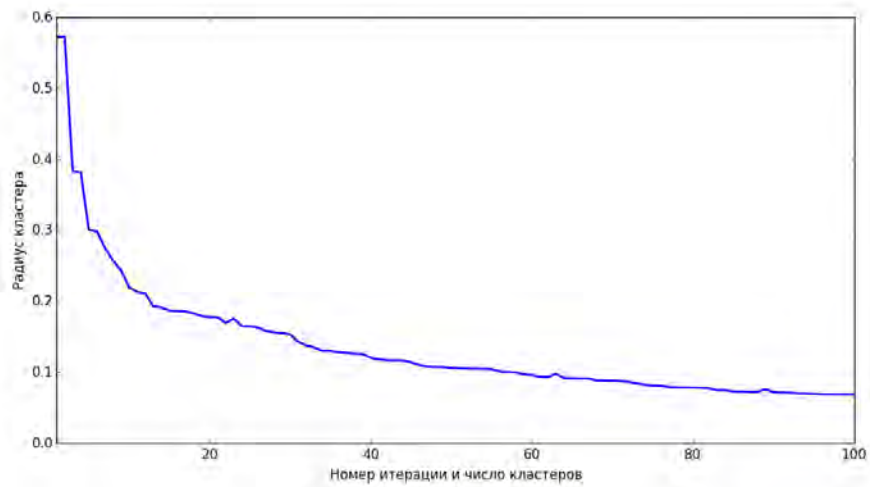
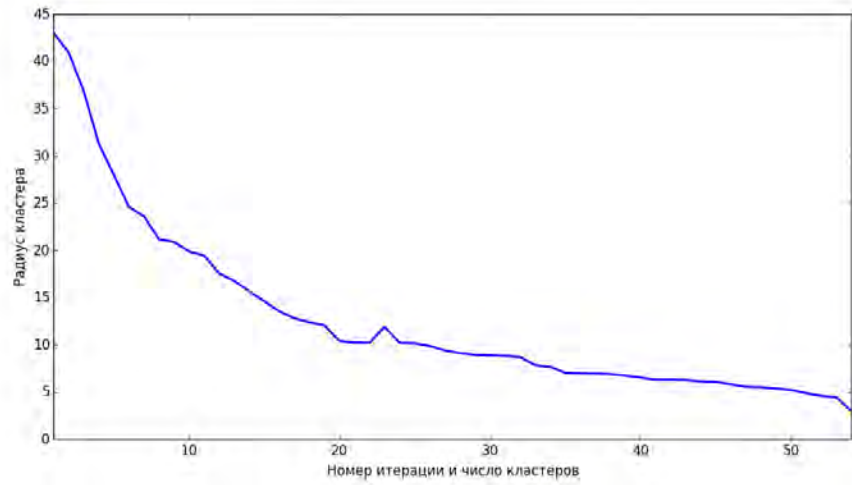


Рис. 4.2: Уменьшение оценки радиуса текущего кластера в течении кластеризации соответственно сверху вниз данных с Regatta, Blue Gene/P (с сетью в 64 процесса) и Ломоносова.

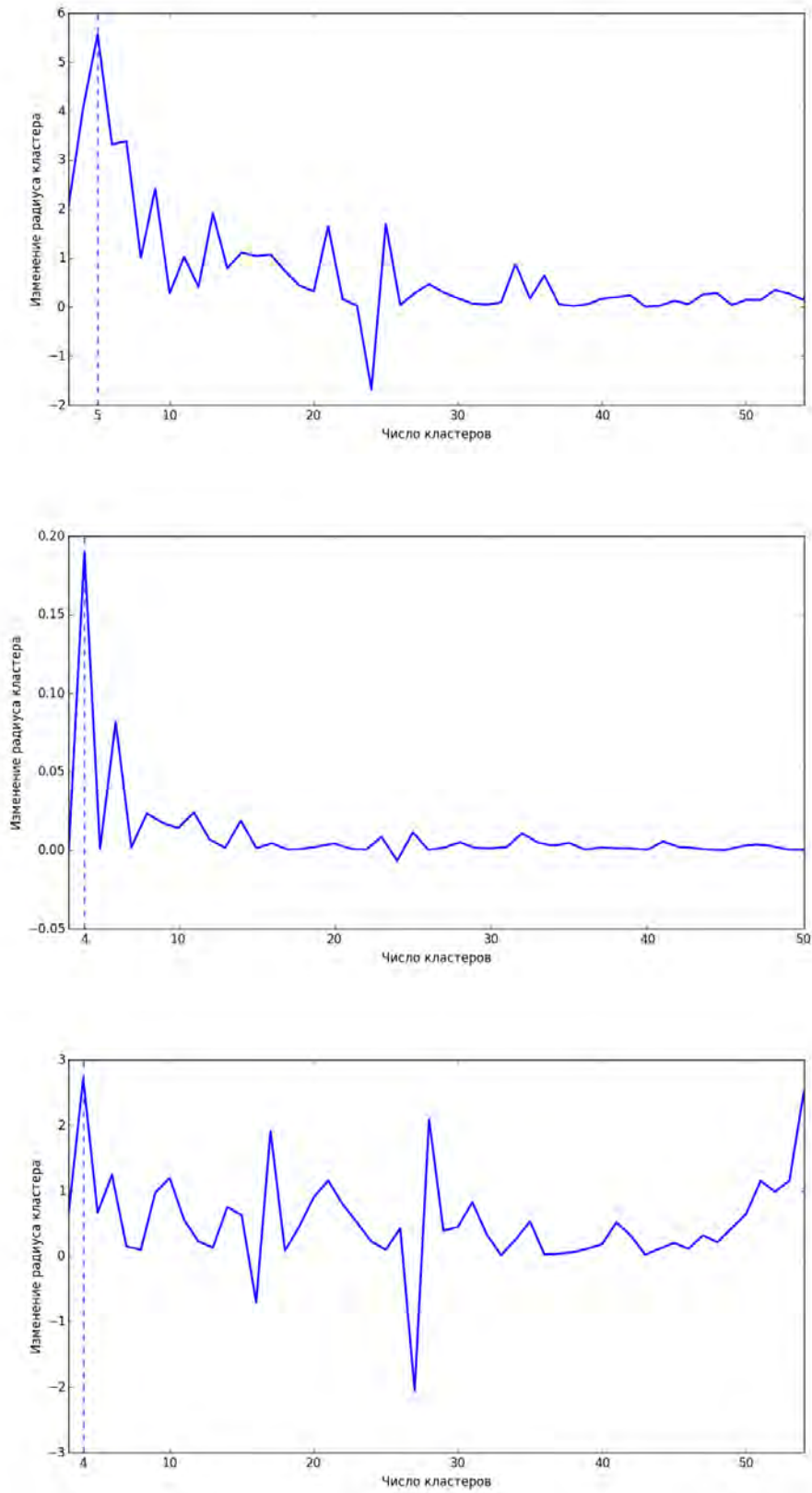


Рис. 4.3: Изменение оценки радиуса текущего кластера на соседних итерациях кластеризации. Соответственно сверху вниз данных с Regatta, Blue Gene/P (с сетью в 64 процесса) и Ломоносова.

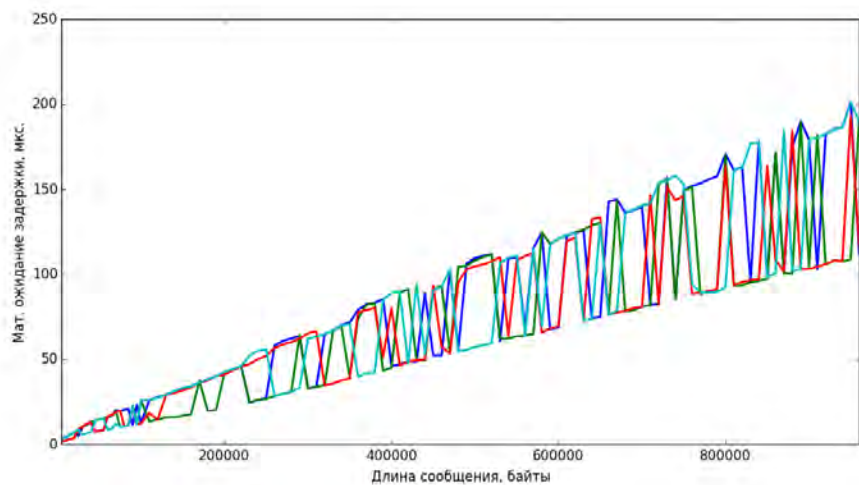
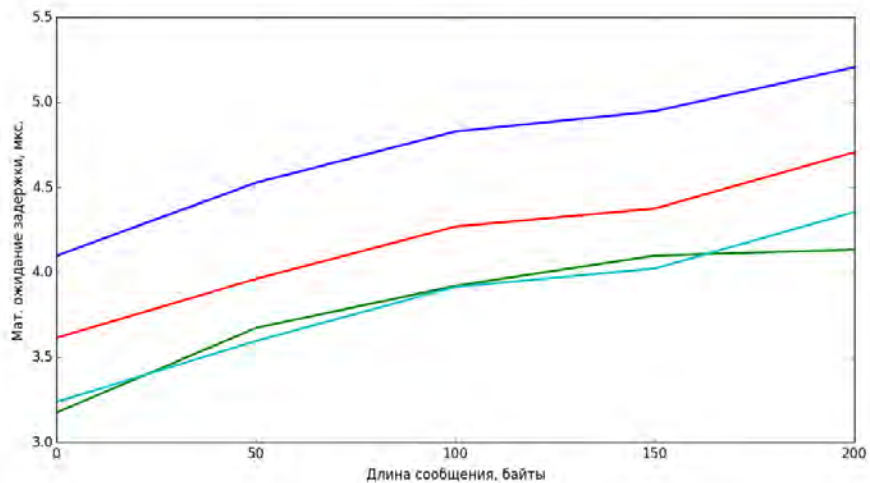
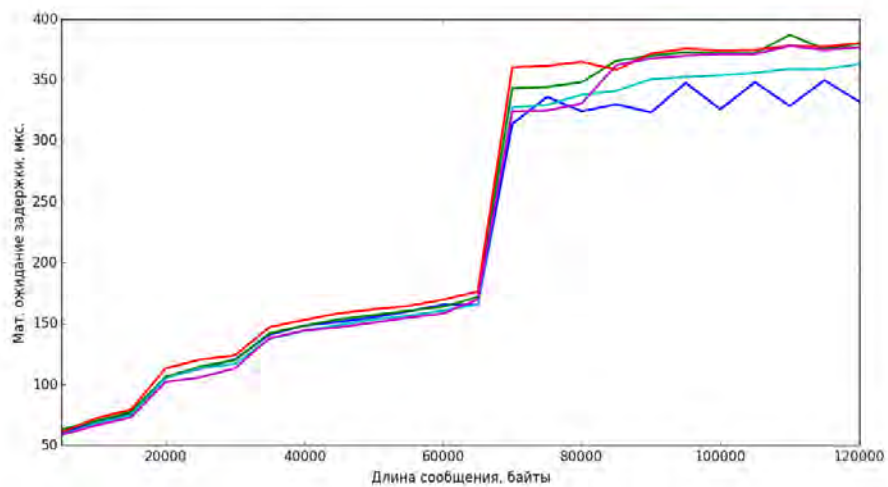


Рис. 4.4: Поведение матожиданий представителей полученных кластеров. На графиках они показаны линиями разных цветом. Соответственно сверху вниз данных с Regatta, Blue Gene/P (с сетью в 64 процесса) и Ломоносова.

Глава 5

Заключение

В данной работе был предложен подход к изучению задержек сообщений в коммуникационной сети вычислительных кластеров, основанный на построении вероятностной модели задержек. Было показано, что предложенный подход позволяет эффективно решать возникающие в предметной области задачи, связанные с анализом задержек. Был предложен метод диагностики коммуникационной сети на основе анализа задержек с использованием построенной модели и был предложен метод агрегирования моделей для отдельных задержек для повышения эффективности их оценки. Рассматриваемые методы были протестированы на реальных данных с суперкомпьютерных систем МГУ им. М. В. Ломоносова.

Список литературы

- [1] Bonnini S. Nonparametric Hypothesis Testing: Rank and Permutation Methods with Applications in R. — 2014.
- [2] Brent R. P. Algorithms for minimization without derivatives. — Courier Corporation, 2013.
- [3] Corlett A., Pullin D., Sargood S. Statistics of one-way internet packet delays // 53rd IETF. — 2002.
- [4] Gallagher M. A., Moore A. H. Robust minimum-distance estimation using the 3-parameter weibull distribution // Reliability, IEEE Transactions on. — 1990. — Vol. 39, no. 5. — P. 575–580.
- [5] Gropp W., Lusk E., Skjellum A. Using MPI: portable parallel programming with the message-passing interface. — MIT press, 1999. — Vol. 1.
- [6] Hill B. M. The three-parameter lognormal distribution and bayesian analysis of a point-source epidemic // Journal of the American Statistical Association. — 1963. — Vol. 58, no. 301. — P. 72–84.
- [7] Johnson S. G. The nlopt nonlinear-optimization package. — 2014.
- [8] Jones E., Oliphant T., Peterson P. {SciPy}: Open source scientific tools for {Python}. — 2014.
- [9] Karakaş M. Determination of Network Delay Distribution over the Internet : Ph. D. thesis / Mehmet Karakaş ; MIDDLE EAST TECHNICAL UNIVERSITY. — 2003.
- [10] Mukherjee A. On the dynamics and significance of low frequency components of internet load // Technical Reports (CIS). — 1992. — P. 300.

- [11] On convergence problems of the em algorithm for finite gaussian mixtures. / Cédric Archambeau, John Aldo Lee, Michel Verleysen et al. // ESANN. — Vol. 3. — 2003. — P. 99–106.

- [12] Salnikov A. N. Parus: A parallel programming framework for heterogeneous multiprocessor systems // Recent Advances in Parallel Virtual Machine and Message Passing Interface. — Springer, 2006. — P. 408–409.

- [13] Voevodin V., Voevodin V. V. Parallel computations. — 2002.