

Оглавление

1-Введение...	2
2-Объекты Сетевого Исчисления...	3-4
3-Основа сетевого исчисления...	4-5
4-Теория очередей и сетевое исчисление...	5-6
5-История создания...	6-7
6-Базовые основы SNC...	7
7-Анализ гарантийного обслуживания...	7-8
8-Базовые основы DNC...	8-9
9-VPN...	9
10-Функции и операции NC...	10
11-Модель системы сетевого исчисления...	10
12-Отличия сетевого исчисления...	10-11
13-Анализ сетевого исчисления...	11
14-Анализ алгебраического сетевого исчисления...	11-14
15-Двойственность исчисления сетей MIN-PLUS и MAX-PLUS...	15-16
16-Вероятностное сетевое исчисление...	16
17-Циклический перебор дефицита (DRR)...	16-17
18-Взвешенный циклический перебор (WRR)...	17-19
19-Библиотека Nansu...	19-23
20-Выводы..	23-224
21-Список используемы...	24-26

1 Введение

Ученые и практики по всему миру по-разному трактуют понятие сетевого исчисления. В статьях и препринтах приводятся следующие определения: Сетевое исчисление — это набор математических результатов, полученных в результате углубленного изучения сетей связи. Это инструмент анализа производительности сети, основанный на алгебре min-plus и алгебре max-plus. Сетевое исчисление вычисляет параметры производительности сети по кривой прибытия и кривой обслуживания, обеспечивая теоретическую основу для гарантии качества потокового мультимедиа в SDN.[1] Сетевое исчисление Network Calculus представляет собой теоретическую основу для определения граничных оценок, параметров качества, обслуживания заявок в пакетных сетях. Основы сетевого исчисления берут свое начало из математической теории диоидов (диоид — алгебра с двумя операциями), в частности, «мин-плюс / макс-плюс» диоида.[2] Сетевое исчисление — это теория систем min-plus, которая позволяет эффективно выводить детерминистические оценки производительности для сетей очередей.[3]

Сетевое исчисление нестатистический метод анализа детерминистических систем, представляет собой альтернативный подход к оценке характеристик потоков данных между компонентами компьютерных сетей. Этот метод основан на алгебре min-plus: во многих случаях он позволяет рассматривать линейные системы, которые являются нелинейными в обычной алгебре.[4]

Теория сетевого исчисления направлена на оценку наихудших характеристик коммуникационных сетей. Он предоставляет методы для анализа моделей, в которых трафик и услуги ограничены некоторыми минимальными и/или максимальными огибающими (кривыми прибытия/обслуживания). Несмотря на появление новых приложений, остается нерешенной сложная и неизбежная проблема: обеспечение точного анализа сетей с помощью агрегированного мультиплексирования.[5] С момента создания теории сетевого анализа были разработаны два её направления - детерминистическое и стохастическое. Теория детерминистического сетевого исчисления DNC (Deterministic NC) [6] применяется при исследовании сетей, предоставляющих гарантии детерминистического обслуживания для поступающих потоков заявок. Для анализа и обеспечения гарантий качества обслуживания в инфокоммуникационных системах и сетях в наибольшей степени подходит теория стохастического сетевого исчисления SNC (Stochastic NC) [7].

Несмотря на появление новых приложений, остается нерешенной сложная и неизбежная проблема: обеспечение точного анализа сетей с помощью агрегированного мультиплексирования. Теория предлагает эффективные методы для ограничения максимальных сквозных или локальных задержек. Однако, как показано в статье [24], эти границы могут быть далеки от точных значений worst case. На данный момент были тщательно проанализированы только сеть с тремя потоками и тремя серверами, а также тандемная сеть под названием sink tree.

2 Объекты Сетевого Исчисления

В сетевом исчислении необходимо различать два вида объектов: реальные перемещения данных и ограничения, которым удовлетворяют эти перемещения. Реальные перемещения данных моделируются кумулятивными функциями: кумулятивная функция $f(t)$ подсчитывает общий объем данных, которые достигли некоторого состояния до времени t (например, общий объем данных, прошедших через данное место в сети). В [5] рассматривают текущую модель, в которой время непрерывно, данные могут быть разделены на произвольно малые фрагменты (измерения времени и данных принадлежат R^+), а кумулятивные функции будут принадлежать $F = \{f : R^+ \rightarrow R^+ \mid f \text{ неубывающий, непрерывный слева, } f(0) = 0\}$. Функции ограничений либо формируют трафик (кривые прибытия), либо гарантируют некоторое обслуживание локально или глобально (кривые обслуживания). Функции ограничения обычно допускают значение $+$. Помимо обычных операций, таких как минимум или сложение функций, сетевое исчисление использует несколько классических ($\min, +$) операций, таких как: пусть $f, g \in F, \forall t \in R^+, \bullet$ **свертка**: $f * g(t) = \inf_{0 \leq s \leq t} (f(s) + g(t-s))$; \bullet **деконволюция**: $(fg)(t) = \sup_{u \geq 0} (f(t-u)g(u))$. Система ввода/вывода представляет собой подмножество S из $(Fin, Fout) \in F \times F \mid Fin \geq Fout$. Он моделирует поток, пересекающий систему, где $F in$ (соотв. $F out$) является кумулятивной функцией на входе (соответственно. выход) системы и $Fin \geq Fout$ указывает на то, что система передает только данные. Система может варьироваться от одного сервера до большой сети с перекрестным трафиком. Траектория системы S является элементом $(F in, F out) \in S$. Система может допускать несколько траекторий $(F in, F out)$, с одним и тем же входом $F in$, связанным с различными выходами $F out$. Учитывая поток данных, пусть $F \in F, \dots, F(t)^-, t, F(0) = 0, \alpha \in F$ является кривой прихода для F , если $\forall s, t \in R^+, s \leq t$, мы имеем $F(t) - F(s) \leq \alpha(t-s)$. F ограничено сверху α . Это означает, что количество битов, поступающих между временем s и t , составляет не более $\alpha(t-s)$. Типичным примером кривой прибытия является аффинная функция $\alpha\sigma, (t) = \sigma + pt, \in R^+$ (иногда называемая кривой прибытия с утечкой), где σ представляет максимальное количество пакетов, которые могут поступать одновременно (максимальный пакет), а p - максимальное время ожидания, срочный показатель прибытий. Обычное предположение состоит в том, что α является субаддитивным (в противном случае его можно заменить его субаддитивным замыканием). Субаддитивное замыкание нашей кривой с протекающим ведром α, σ, ρ по-прежнему остается $\alpha\sigma, \rho$, за исключением $t = 0$, где значение должно быть равно 0. Сеть будет моделироваться, без потери общности, ориентированным графом, где потоки должны следовать по краям, а серверы (коммутаторы, линии передачи, маршрутизаторы ...) представлены вершинами.

Серверы и потоки будут идентифицироваться по индексам. Набор серверов равен $S = \{1, \dots, n\}$, и каждый сервер j предлагает строгую кривую обслуживания $\in F, -(|\beta_j| \text{ частей})$ и выпуклой. Таким образом, можно записать $\beta_j(t) = \max_{1 \leq i \leq |\beta_j|} (r_j, t^i h, i) r_j, i, h_j, i \in R^+$. Множество потоков

равно $F = \{1, \dots, p\}$. Каждому потоку i соответствует пара (a_i, μ_i) , где μ_i - (конечная) упорядоченная последовательность серверов, пересекаемых потоком, а a_i - кривая прибытия для кумулятивной функции перед входом на первый сервер. Предполагается, что a_i неотрицательный, неубывающий, кусочно-аффинный (с конечным числом $|a_i|$ частей) и вогнутый.

3 Основа сетевого исчисления.

Основой сетевого исчисления являются свертка min-plus и свертка min-plus, где операции min-plus вытекают из соответствующих классических операций, заменяя сложение вычислением минимума и умножение сложением [4]. Операции свертки min-plus и свертки двух функций.

Определение 1 (Свертка с минусом и свертка без свертки). Минимальная свертка двух функций $x(t)$ и $y(t)$ определяется как

$$z(t) = (x \oplus y)(t) = \inf_{s \in [0, t]} [x(ts) + y(s)]$$

и минимальная плюс де-свертка в виде

$$z(t) = (x \odot y)(t) = \sup_{s \in [0, \infty)} [x(t+s) - y(s)].$$

Две величины, а именно отставание и задержка, как определено ниже, представляют особый интерес в контексте сетей и сетевых приложений.

Определение 2 (Отставание и задержка). Рассмотрим сетевой элемент без потерь с входом $F(t)$ и выходом $F'(t)$. Отставание в момент времени u равно $b(u) = F(u) - F'(u)$.

Предположим, что данные подаются в последовательности поступления. Задержка в момент времени u равна $d(u) = \inf\{t \geq 0 : F(u) \leq F'(u+t)\}$.

В контексте сетевого исчисления кривые прибытия и обслуживания играют важную роль в характеристике прибытия трафика и предлагаемых услуг. Кривые прибытия представляют собой верхние границы входных и выходных данных элементов сети. Нижняя граница услуги, предоставляемой сетевым элементом, например планировщиком на исходящей линии связи, может характеризоваться кривой минимального обслуживания, также называемой кривой обслуживания.

Определение 3 (Кривые прибытия и обслуживания). Возрастающая функция $\alpha(t) \subset \alpha(t) = 0$ при $t \leq 0$ представляет собой кривую прибытия для транспортного потока с кумулятивной функцией прибытия $F(t)$, если для всех $t \geq 0$ для всех $u \geq t$ $\alpha(t) \geq F(u) - F(u)$.

Сетевой элемент с входной функцией прибытия $F(t)$ и выходной функцией прибытия $F'(t)$ предлагает кривую обслуживания $\beta(t) \subset \beta(t) = 0$ для $t \leq 0$, если для всех $t \geq 0$ $F'(t) \geq (F \oplus \beta)(t)$.

Характерным типом кривой поступления является ограничение на негерметичность ведра $\alpha(t) = b + rt$ при $t > 0$, которое допускает определенный размер пакета b и устойчивую скорость r . Различные дисциплины планирования, включая приоритетную очередь и обобщенное совместное использование процессоров, могут быть описаны кривыми обслуживания типа скорость-задержка $\beta(t) = R[t - T]^+$, где $[...]^+ = 0$, если аргумент отрицательный. Следствие: (Мультиплексирование). Рассмотрим два потока

с кривыми прибытия $\alpha_1(t)$ и $\alpha_2(t)$, которые мультиплексируются. Кривая прибытия для совокупности трафика равна $\alpha_{1,2}(t) = \alpha_1(t) + \alpha_2(t)$. Обратите внимание, что детерминистическое мультиплексирование учитывает совместный наихудший случай, то есть наиболее неблагоприятную суперпозицию поступающего трафика.

Теорема 1 (Конкатенация). Рассмотрим два узла с кривыми обслуживания $\beta_1(t)$ и $\beta_2(t)$ последовательно. Объединенная система предлагает кривую обслуживания $\beta_{1,2}(t) = (\beta_1 \otimes \beta_2)(t)$. Объединенная кривая обслуживания для более чем двух узлов немедленно следует за рекурсивным приложением. Важность теоремы конкатенации обусловлена линейным масштабированием числа узлов, которое достигается для объединенных кривых обслуживания, а также для последовательных границ производительности.

Теорема 2 (Отставание, задержка и границы выходных данных). Рассмотрим узел с кривой обслуживания $\beta(t)$ с входом, который ограничен кривой прибытия $\alpha(t)$. Граница отставания b задается через $b = (\alpha \oslash \beta)(0)$.

Предполагая, что при планировании первого входа первым выходом граница задержки d равна $d = \text{INF}[t \geq 0 : (\alpha \oslash \beta)(-t) \leq 0]$.

Граница выходного сигнала $\alpha'(t)$ задается через: $\alpha'(t) = (\alpha \oslash \beta)(t)$.

Поток NC представлен функцией времени $A(t)$, которая подсчитывает количество трафика, поступившего ко времени t . Такая функция обязательно является расширяющейся в широком смысле. Часто предполагается, что он непрерывен слева направо, т.е. $A(t)$ представляет количество битов в $[0, t]$. В частности, $A(0) = 0$. Потоки могут быть ограничены кривыми прибытия. Функция увеличения α в широком смысле представляет собой кривую прихода для потока A , если

$$A(t) - A(s) \leq \alpha(t - s), \forall s \leq t$$

Например, формирователь негерметичного ведра со скоростью ρ и размером разрыва σ обеспечивает вогнутую аффинную кривую поступления $\gamma\sigma, \rho(t) = \rho + pt$. В частности, это означает, что долгосрочная скорость поступления потока не может превышать ρ . Формирователи "негерметичных" пакетов часто используются на входе в сеть, чтобы гарантировать, что вводимый трафик не превысит согласованный объем. $A(t)$ всегда находится ниже кривой прибытия $a(t)$ и ее перемещений вдоль $A(t)$.

4 Теория очередей и сетевое исчисление

Явления очередей и задержек в современной науке считаются случайными и описываются стохастическими моделями. Изучением таких моделей занимается обширный раздел теории вероятностей – **теория очередей (Queueing theory)** или теория массового обслуживания. Рождение тео-

рии очередей относится к началу 20 в. и обязано изобретению и широкому распространению телефона. Исторически первая задача теории очередей была связана с расчетом необходимого количества телефонных линий при автоматической коммутации вызовов. Было показано, что не существует единственного абсолютно правильного решения – решение имеет вероятностный характер. Впоследствии было изучено много различных систем с очередями и найдены их средние статистические характеристики (времена ожидания, длины очередей и др.). В конце 20 в. в связи с бурным развитием компьютерных сетей и особенно сети Интернет возник новый подход – рассматривать очереди и задержки как детерминированные явления, работая не с вероятностными характеристиками, а с регулярными ограничениями. Соответствующий раздел прикладной математики получил название сетевого исчисления (Network calculus).

В отличие от теории очередей сетевое исчисление предлагает более прозрачную методику решения для сложных систем, работает с неопределенными входящими потоками и позволяет рассчитывать наихудшие значения прогнозируемых величин. Основную идею сетевого исчисления можно обозначить следующим образом: надейся на лучшее, но ожидай худшего. В ряде случаев, особенно, для объектов повышенной опасности, сетевое исчисление оказывается предпочтительнее теории очередей.

5 История создания

Основателем теории НС является профессор Калифорнийского университета Р.Л. Круз (1959-2013), опубликовавший в 1991 году две части своей знаменитой статьи. В первой части статьи рассматриваются характеристики различных сетевых элементов, таких как, линия с постоянной задержкой, входной буфер, мультиплексор, демультиплексор и т.д. Основное внимание в этой и последующих работах Круза, а также его последователей было уделено (σ, ρ) -контроллеру трафика, аналогичному алгоритму "bucket где параметр ρ характеризует скорость входного транспортного потока, а σ - его всплеск. Причина в том, что этот регулятор наиболее подходит для описания трафика современных пакетных сетей. Основная идея предлагаемого способа анализа функционирования пакетной сети заключалась в определении взаимосвязи между границами пакетности потоков трафика и характеристиками функционирования узлов в различных точках сети. Очевидно, что если входящий (σ, ρ) -трафик обслуживается сетевым узлом с пропускной способностью ρ , то количество пакетов в узле (отставание) никогда не будет больше σ . С момента создания теории сетевого анализа были развиты два ее направления: детерминистическое и стохастическое. Теория детерминистического НС (DNC) используется при изучении сетей, обеспечивающих детерминистические гарантии обслуживания для входящих потоков пакетов. DNC может использоваться для исследования систем, которые получают неизвестный трафик, но имеют детерминистические ограничения на максимальное количество принимаемых пакетов, а обслуживание

входящих пакетов в системе также осуществляется с детерминистическими верхними границами производительности. К таким системам относятся локальные компьютерные сети, специализированные бортовые сети (например, авиационные), системы на чипе и т.д. детерминистические модели NC, в отличие от стохастических моделей, проще и понятнее для восприятия. Однако часто на практике процессы приема и обслуживания приложений в системе характеризуются стохастическим процессом, и требуется определенная вероятностная гарантия QoS. Примером могут служить современные мультисервисные сети связи, которые обслуживают мультимедийные потоки голоса, видео и пакетов данных.

6 Базовые основы SNC

Стохастическое сетевое исчисление (SNC) является расширением сетевого исчисления, которое учитывает статистический выигрыш от мультиплексирования путем замены детерминистических границ статистическими. детерминистическая огибающая прибытия не может быть нарушена при любых s и t , где $0 \leq s \leq t$. Расширение огибающей прибытия в вероятностном режиме ослаблено таким образом, что они допускают небольшую вероятность нарушения. Статистические конверты прибытия имеют много вариаций в литературе. Границы определяются либо как последовательность случайных величин, либо с помощью детерминистических функций. Первые границы представляют собой случайные процессы, которые стохастически больше, чем соответствующие процессы трафика на любом временном интервале. По сравнению с классической теорией массового обслуживания, стохастическое сетевое исчисление включает в себя гораздо больше случайных процессов, включая зависящий от большого расстояния, самоподобный и сильно зависящий трафик. Эта общность достигается за счет точных решений. Вместо этого стохастическое сетевое исчисление вычисляет неасимптотические статистические оценки производительности типа $P[\text{отставание} > x] \leq P[\text{отставание} > x] \leq \dots, 10 - 6, \dots, (MGFS), \dots, MGFS, \dots, \dots, \dots, \dots, \dots$.

7 Анализ гарантийного обслуживания

В контексте сетевого исчисления для детерминистического анализа гарантии обслуживания (детерминистическая) модель трафика кривой прибытия и (детерминистическая) модель сервера кривой обслуживания являются наиболее фундаментальными и важными концепциями. Их определения заключаются в следующем:

Определение 1. Поток имеет (детерминистическую) кривую прибытия $\alpha \in F$, если для всех $0 \leq s \leq t$ выполняется: $A(s, t) \leq \alpha(t - s)$.

Определение 2. Сервер обеспечивает (детерминистическую) кривую обслуживания $\beta \in F$ для своего прибытия A , если для всех $t \geq 0$ его отправление $A * \beta$ удовлетворяет: $A * (t) \geq A \otimes \beta(t)$.

Определение 3. Поток имеет стохастическую кривую прибытия, ориентированную на количество трафика (т.е. с), $\alpha \in F$ с ограничивающей функцией $f \in F$, обозначаемой $A \sim ta < f, \alpha >$, если для всех $0 \leq s \leq t$ и всех $x \geq 0$ выполняется: $P\{A(s, t) - \alpha(ts) > x\} \leq f(x)$.

Определение 4. Поток имеет виртуально - центрическую (v. b. c.) стохастическую кривую прибытия $\alpha \in F$ с ограничивающей функцией $f \in F$, обозначаемой $A \sim vb < f, \alpha >$, если для всех $t \geq 0$ и всех $x \geq 0$ выполняется: $P\{SUP 0 \leq s \leq t[A(s, t) - \alpha(ts)] > x\} \leq f(x)$.

Большинство серверных моделей относятся к слабой стохастической кривой обслуживания X:

Определение 5. Сервер S доказывает слабую стохастическую кривую обслуживания $\beta \in F$ с ограничивающей функцией $g \in F$, обозначаемой $S \sim ws < g, \beta >$, если для всех $t \geq 0$ и всех $x \geq 0$, выполняется $P\{A \otimes \beta(t) - A * (t) > x\} \leq g(x)$.

8 Базовые основы DNC

Пусть поток пакетов прибытия, поступающий на узел, характеризуется кумулятивной функцией (кривой) $A(t)$. Поток отправляющихся пакетов на выходе узла характеризуется кумулятивной функцией (кривой) $D(t)$. Основная идея теории детерминистического сетевого исчисления (DNC) заключается в том, что граничные значения характеристик QoS могут быть получены путем определения максимальных горизонтальных и вертикальных расстояний между кривыми детерминистических функций $A(t)$ и $D(t)$. В частности, значения задержки и отставания определяются, соответственно, как:

$$\delta(A, D) := SUP(INF[d \in R^+, D(t + d) \geq A(t)])$$

$$t \geq 0$$

$$b(A, D) := SUP(A(t) - D(t))$$

$$t \geq 0$$

Для вычисления граничных значений характеристик QoS необходимо указать законы распределения, описывающие поток поступающих пакетов и дисциплину его обслуживания в узле. Для этого NC использует кривую прибытия $\alpha(t)$, описывающую поток трафика, поступающий на узел, и кривую обслуживания $\beta(t)$, описывающую дисциплину обслуживания в узле. Функция будет представлять собой кривую прибытия потока трафика на входе узла при условии, что в течение некоторого наблюдаемого периода времени $[0, t]$ выполняется следующее неравенство:

$$A(t) - A(s) \leq \alpha(t - s), 0 \leq s \leq t$$

Следовательно, фактическое количество пакетов прибытия в потоке всегда будет меньше или равно количеству пакетов, определяемому на основе кривой прибытия, т.е. оно определяет наименьшую верхнюю границу количества пакетов в потоке пакетов прибытия. Модель кривой обслуживания описывает сетевые элементы, такие как маршрутизаторы, планировщики и каналы связи, с использованием функций времени, которые определяют

услугу, предлагаемую элементом в течение определенного интервала времени. Подобно теории систем, кривые обслуживания можно рассматривать как импульсную характеристику линейной системы, однако в рамках другой, так называемой алгебры мин-плюс (также известной как тропическая алгебра). В этой алгебре отправление данных сетевого элемента может быть вычислено на основе свертки поступающих данных и кривой обслуживания системы. Эта форма свертки важна, поскольку она устанавливает общую основу для анализа целых сетей. Отдельные системы вдоль сетевого пути могут быть легко объединены путем свертки их кривых обслуживания, что дает кривую сетевого обслуживания, которая определяет сквозную доступную услугу. Функция $\beta(t)$ представляет собой кривую обслуживания для узла сети при условии, что в течение некоторого наблюдаемого периода времени $[0, t]$: $D(t) \geq \text{INF}[(t-s) + (s)] = (\otimes \beta)(t), 0 \leq s \leq t'$

$\otimes - (\min, +)$ -свертка. Таким образом, граничные значения задержки пакетного трафика и отставания узла будут определяться соответственно как максимальные горизонтальные и вертикальные расстояния между кривыми $\alpha(t)$ и $\beta(t)$. Наиболее важным свойством DNC является расширение концепции кривой обслуживания от одного узла до произвольного числа узлов сети, соединенных последовательно. Это позволяет получать сквозные характеристики всей сети, используя методику оценки граничных характеристик для одного узла

9 VPN

Теория NC была использована в рекомендации ITU-T Y.1315 [5] для моделирования входящего потока и работы узла VPN. На основе этого подхода в [6] было предложено использовать теорию NC также для решения задач оптимального планирования VPN. Модели сети VPN на основе DNC и SNC могут быть совместно использованы для определения границ сквозных характеристик QoS при определении оператором оптимального маршрута на этапе планирования VPN. DNC можно использовать в случае, если клиенту необходимы строгие гарантии QoS и нарушение их очень критично (например, в корпоративных сетях банков, сетях связи спецслужб). При этом оператору необходимо обеспечить дополнительное резервирование сетевых ресурсов, что естественно приведет к увеличению стоимости предоставляемой услуги. SNC подойдет в случае, если клиенту не важны нарушения гарантий QoS (например, если клиент планирует использовать VPN сеть для передачи не критичных данных, для выхода в сеть Интернет и т.п.). Алгоритмы определения границ сквозных характеристик QoS на основе моделей сети VPN, построенные с использованием теорий DNC и SNC, могут быть реализованы в пакете прикладных программ, который позволит автоматизировать процесс планирования услуг VPN.

10 Функции и операции НС

Основной целью сетевого исчисления является анализ производительности сетей связи. Потоки и услуги в сети моделируются с помощью неубывающих функций $t \mapsto f(t)$, где t - время, а $f(t)$ - объем данных. Существуют различные модели в зависимости от того, принимает ли t (соответственно $f(t)$) дискретные или непрерывные значения, например, в \mathbb{N} или \mathbb{R}^+ . В сетевом исчислении необходимо различать два вида объектов: реальные перемещения данных и ограничения, которым удовлетворяют эти перемещения. Реальные перемещения данных в основном моделируются кумулятивными функциями: кумулятивная функция $f(t)$ подсчитывает общий объем данных, которые достигли некоторого состояния до времени t (например, общий объем данных, прошедших через данное место в сети). Кумулятивные функции являются левосторонними. Это не является большим ограничением для разработчика моделей. Это предположение, тем не менее, имеет техническое значение в системе сетевого исчисления (например, при определении начала периодов задержки). Напротив, к функциям ограничения не предъявляется никаких предположений о непрерывности (слева или справа).

11 Модель системы сетевого исчисления

НС моделирует сеть как ориентированный граф, называемый граф серверов. Сервер предлагает буфер для постановки в очередь входящего запроса (данных) и служебный ресурс (пересылка данных). Данные вводятся в сеть потоками. Предполагаются одноадресные потоки с одним исходным сервером и одним приемным сервером, а также фиксированный маршрут между ними. Потребность в пересылке потоков характеризуется функциями, кумулятивно подсчитывающими их данные,
$$F_{16}^+ = \{g : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \mid g(0) = 0, \forall s \leq t : g(t) \geq g(s)\}$$

12 Отличия сетевого исчисления.

Важное отличие сетевого исчисления от классических, более распространенных и отработанных методик теории очередей – работа с детерминистическими (регулярными) ограничениями на потоки и на обслуживание, а не с их стохастическими моделями. Поэтому и результаты автоматически получаются в виде ограничений (максимальных значений). Работа с ограничениями имеет ряд преимуществ. Детерминистические ограничения на входящий поток позволяют работать с неопределенным входящим потоком. Это оказывается полезным в случае, если распределение входящего потока неизвестно или имеет сложный вид. Процедуры обслуживания, выраженные в виде ограничений, более удобны в использовании. Ограничения на результаты предпочтительнее, поскольку для дальнейших планов и расчетов требуются характеристики детерминистические, а статистические

характеристики неудобны в использовании. Одно из наиболее важных преимуществ работы с ограничениями заключается в том, что ограничения на результаты, т.е. наихудшие значения прогнозируемых величин, представляют наибольший интерес для промышленных объектов повышенной опасности (например, объектов атомной энергетики). В настоящее время сетевое исчисление широко используется для расчета характеристик больших и малых вычислительных сетей, при этом особое значение сетевое исчисление имеет для систем повышенного риска эксплуатации. Кроме того, наряду с применением по своему прямому назначению для вычислительных сетей, в последнее время сетевое исчисление стало применяться для объектов повышенной опасности при решении других, не связанных с сетями задач. Одна из таких задач – оценка максимального времени модификации баз данных [26], когда необходимо предсказать не среднее время модификации, а максимальное (гарантированное).

13 Анализ сетевого исчисления

Анализ сетевого исчисления (NC) использует простую модель, состоящую из сети планировщиков и пересекающих их потоков данных. Затем можно применить ряд аналитических “строительных блоков” для построения модели, не навязывая пессимистических предположений, таких как самосогласование на тандемах серверов. Тем не менее, добавления пессимизма не всегда можно избежать. Таким образом, вычисление наилучшей оценки сквозной задержки отдельного потока сводится к поиску наименее пессимистичных моделей конкуренции для всех тандемов планировщиков в сети – и исчерпывающий поиск может легко стать очень ресурсоемкой задачей. В литературе предлагается многообещающее решение этой дилеммы: эвристика, использующая предсказания машинного обучения (ML) внутри анализа NC. Хотя результаты этой работы были многообещающими с точки зрения качества, связанного с задержкой, и вычислительных затрат, практически нет понимания того, когда делается прогноз или может ли обученный алгоритм достичь столь же поразительных результатов в сетях, сильно отличающихся от его обучающих данных.

14 Анализ алгебраического сетевого исчисления

Анализ NC направлен на вычисление оценки наихудшей задержки, с которой сталкивается конкретный поток интересов на своем пути. Алгебраический анализ делает это, выводя (минимальный, плюс)-алгебраический член, который ограничивает задержку. Кривые обслуживания на пути являются общими для всех потоков, пересекающих соответствующий сервер, но кривая прибытия известна только на исходном сервере соответствующего потока. Чтобы вывести из такой модели конечную задержку интересующе-

го потока, анализ NC основан на (\min, plus) -алгебраических манипуляциях с кривой.

Определение 4 (Операции с ЧПУ): (минимум, плюс)-алгебраическая агрегация, свертка и деконволюция двух функций $g, h \in F0$ определяются как

$$\text{aggregation: } (g + h)(d) = g(d) + h(d), \quad (4)$$

$$\text{convolution: } (g \otimes h)(d) = \inf_{0 \leq u \leq d} \{g(d-u) + h(u)\}, \quad (5)$$

$$\text{deconvolution: } (g \oslash h)(d) = \sup_{u \geq 0} \{g(d+u) - h(u)\}. \quad (6)$$

Объединение кривых прибытия создает единую кривую прибытия для их мультиплекса. С помощью свертки тандем серверов можно рассматривать как единый сервер, обеспечивающий единую кривую обслуживания. Деконволюция позволяет вычислить кривую прибытия, ограничивающую поток (или совокупность потоков). $A'(t)$ после сервера. Отставание и задержка могут быть ограничены следующим образом: **Теорема 3(Границы производительности):** рассмотрим сервер s , который предлагает кривую обслуживания β . Предполагаем, что поток f с кривой прибытия α пересекает сервер. Тогда получаются следующие оценки производительности для $f, \forall t \in R+$:

$$\text{output: } \alpha'(t) = (\alpha \oslash \beta)(t) \quad (7)$$

$$\text{backlog: } B(t) \leq (\alpha \oslash \beta)(0) = vDev(\alpha, \beta) \quad (8)$$

$$\begin{aligned} \text{delay: } D(t) &\leq \inf \{d \geq 0 \mid (\alpha \oslash \beta)(-d) \leq 0\} \\ &= hDev(\alpha, \beta) \end{aligned} \quad (9)$$

Учитывая кривые β , ограничивающие нижнюю границу доступной услуги пересылки, и α , ограничивающие верхнюю границу поступающих данных, NC может вычислить нижние границы остаточного обслуживания интересующего потока. **Теорема 4(Кривые остаточного обслуживания для мультиплексирования FIFO):** Пусть поток f имеет кривую прибытия α . Гарантированное остаточное обслуживание на сервере FIFO s , обслуживающем f , ограничено бесконечным набором кривых обслуживания

$$\begin{aligned} \beta_{\theta, \alpha}(t) &= [\beta(t) - \alpha(t - \theta)]^{\uparrow} \cdot \mathbf{1}_{\{t > \theta\}} \\ &=: \beta \ominus_{\theta} \alpha, \quad \forall \theta \geq 0 \end{aligned} \quad (10)$$

, где $\mathbf{1}_{\{\text{условие}\}}$ обозначает тестовую функцию (1, если условие истинно, 0 в

противном случае), а $[g(x)] \uparrow = \sup 0 \leq z \leq xg(z)$ - неубывающее замыкание $g(x)$, определенное при положительных действительных значениях $x \in R + 0$. Свободный параметр отражает потенциальное влияние мультиплексирования FIFO без необходимости уже знать другие потоки в s . Он определяет бесконечный набор (допустимых) кривых остаточного обслуживания при наличии f . Наилучшая настройка \emptyset зависит от кривой поступления потока процентов, обслуживаемой $\beta\emptyset, \alpha$, а также от характеристики производительности, которая должна быть ограничена – задержка потока процентов, отставание или выход. Анализ NC FIFO: Анализ NC FIFO широко использует теорему 4, ставя задачу установить все взаимозависимые параметры для наилучшей возможной границы. Анализ наименьшей верхней границы задержки, преобразует алгебраический член NC с его параметрами в формулировку LP. Преобразование ограничено обслуживанием корзины токенов и поступлениями с задержкой скорости. Первоначально, Наименьшая Верхняя Задержка. Bound сосредоточился на ограничении задержки интересующего потока, т.е. на обеспечении тандемного анализа, в то время как последняя работа включила его в анализ прямой связи с ЧПУ, что привело к анализу Наименьшей верхней границы задержки-FF, что увеличило необходимость оптимизации для ограничения выходных данных. Для простоты и независимости от фактической реализации мы будем использовать Наименьшую верхнюю границу задержки-анализ FF и FIFO.

Вычисление сенсорной сети: Случай с одним приемником Предполагается, что используемый протокол маршрутизации формирует дерево в сенсорной сети. Следовательно, заданы N сенсорных узлов, расположенных в ориентированном ациклическом графе. Каждый сенсорный узел i воспринимает свое окружение и, таким образом, подвергается воздействию входной функции R_i , соответствующей его воспринимаемому входному сигналу. Если узел датчика i не является конечным узлом дерева, то он также получает данные о датчиках от всех своих дочерних узлов $child(i, 1), \dots, child(i, n_i)$, где n_i - количество дочерних узлов узла датчика i . Узел датчика i пересылает/обрабатывает свои входные данные, что приводит к выходной функции $R * i$ от узла i к его родительскому узлу. Теперь необходимо включить основные компоненты сетевого расчета, кривую прибытия и обслуживания. Сначала должна быть получена кривая прибытия a_i каждого сенсорного узла в поле. Входной сигнал каждого сенсорного узла в поле, принимая во внимание его воспринимаемый входной сигнал и его дочерний входной сигнал, равен:

$$\bar{R}_i = R_i + \sum_{j=1}^{n_i} R_{child(i,j)}^* \quad (1)$$

Таким образом, кривая прихода для общей входной функции для сенсорного узла i равна:

$$\bar{\alpha}_i = \alpha_i + \sum_{j=1}^{n_i} \alpha_{child(i,j)}^* \quad (2)$$

Расчет сети датчиков: Случай с несколькими приемниками Базовая настройка, а также процедура расчета могут быть восстановлены из случая с одним приемником. Однако анализ сетевого потока значительно более сложен в случае с несколькими приемниками. Когда анализируется конкретный поток, представляющий интерес, между данным источником и поглотителем, эффект перекрестного потока из других необходимо принимать во внимание потоки в разные стоки. Здесь можно использовать известный результат сетевого исчисления, рассказывающий, как смоделировать кривую обслуживания узла FIFO при перекрестном трафике из других потоков [9, глава 6]

Algorithm 1 Multiple Sink Network Flow Analysis

ComputeOutputBound(from node i , flows \mathbb{F})

```

for all  $\text{pred}(i)$ 
   $\alpha_i^{pred} += \text{ComputeOutputBound}(\text{pred}(i), \{\text{flows to node } i \text{ from } \text{pred}(i)\} \cap \mathbb{F})$ 
   $\alpha_{excl} += \text{ComputeOutputBound}(\text{pred}(i), \{\text{flows to node } i \text{ from } \text{pred}(i)\} \setminus \mathbb{F})$ 
find  $\theta^{opt}$  such that  $\beta_i^{eff} = \max_{\theta \geq 0} \{\beta_i(t) - \alpha_{excl}(t - \theta)\}$  is maximal
return  $\alpha_i^{pred} \oslash \beta_i^{eff}$ 

```

ComputeDelayBound (flow of interest f)

```

ComputeOutputBound(sink of  $f$ ,  $\text{pred}(\text{sink of } f)$ )
 $\text{delay}_{total} = 0.0$ 
for all nodes  $i$  on the path from source to sink of  $f$ 
   $\text{delay}_{total} += h_{\alpha_i^{pred}, \beta_i^{eff}}$ 
return  $\text{delay}_{total}$ 

```

Теорема 5 (Кривая обслуживания узла FIFO) Рассматривается узел FIFO, мультиплексирующий два потока 1 и 2. Предполагается, что узел гарантирует строгую минимальную кривую обслуживания β для совокупности двух потоков. Поток 2 имеет α_2 в качестве кривой прихода. Определим семейство функций $\beta_1 \oslash () = [\beta() \alpha_2()] + 1\{ > \emptyset \}$, то для любого $\emptyset \geq 0$ стока 1 это гарантированный сервис $\beta_1 \oslash \emptyset$ (если $\beta_1 \oslash \emptyset$ имеет широкий смысл увеличивать). Здесь функция индикатора 1_{cond} определяется как 1, если cond равно true, и 0 в противном случае. Теорема 5 теперь позволяет выводить кривые обслуживания, специфичные для приемника, для каждого узла на пути между источником и приемником для определенного потока, представляющего интерес.

15 Двойственность исчисления сетей MIN-PLUS и MAX-PLUS

Сетевое исчисление min-plus с непрерывным временем проводит анализ сетевых элементов в рамках (F_o, \wedge, \otimes) алгебры диоидов, где F_o - множество непрерывных слева неубывающих функций $F : R \rightarrow R + o \cup \{+\infty\}$, при этом $F(t) = 0$, если $t \leq 0$, \wedge -операция является точечным минимумом, а \otimes - свертка с минусом, которая определяется как $F \otimes G(t) = \inf_{0 \leq s \leq t} \{F(s) + G(ts)\}$ для двух функций $F, G \in F_o$. Совокупное количество прибытий и отправлений в элементе сети за интервал времени $[0, t)$ задается $A(t)$ и $D(t)$, соответственно, $A, D \in F_o$. Доступное обслуживание в сетевом элементе выражается в терминах функции $S \in F_o$, называемой кривой минимального обслуживания, которая удовлетворяет $D(t) \geq A \otimes S(t)$ для всех t . Когда прибытия ограничены функцией $E \in F_o$, такой, что $E(s) \geq A(t+s) - A(t)$ для всех s и t , мы говорим, что E является огибающей трафика для A . Функции в сетевом исчислении max-plus вычисляют время прибытия или событие отправления для заданного количества битов. В версии с непрерывным пространством используется диоид (T_o, \vee, \otimes) , где T_o - множество непрерывных справа неубывающих функций $F : R \rightarrow R + o \cup \infty \cup +\infty$, $F(t) = \infty$, $t < 0$, $F(t) \geq 0$, $t \geq 0$. Операция \vee - это точечный максимум, а \otimes - свертка max-plus, при этом $F \otimes G(t) = \sup_{0 \leq s \leq t} \{F(s) + G(t-s)\}$ для двух функций $F, G \in F_o$. Прибытия и отправления описываются функциями $4A \in T_o$ и $TD \in T_o$. Здесь $TA(t)$ - это время поступления бита, где битовые значения могут быть действительными числами. Кривая минимального обслуживания - это функция $yS \in T_o$ такая, что $TD(t) \leq TA \otimes yS(t)$ для всех t , а огибающая трафика $E \in T_o$ для функции времени прибытия удовлетворяет $E(t) \leq TA(t) - TA(0)$ для всех t . Существует изоморфизм между сетевым исчислением min-plus и max-plus с помощью псевдоинверсных функций

$$F \downarrow(y) = \inf\{x | F(x) \geq y\} = \sup\{x | F(x) < y\},$$

$$F \uparrow(y) = \sup\{x | F(x) \leq y\} = \inf\{x | F(x) > y\},$$

где $F \downarrow$ называется нижним псевдоинверсом, а $F \uparrow$ - верхним псевдоинверсом. Псевдоинверсии устанавливают следующие соотношения:

- $F \in F_o \rightarrow F \uparrow \in T_o$.

- $F \in T_o \rightarrow F \downarrow \in F_o$.

$F \downarrow$ является непрерывным слева, а $F \uparrow$ является непрерывным справа.

- F является левосторонним непрерывным $\rightarrow F = (F \uparrow) \downarrow$.

- F является прямым непрерывным $\rightarrow F = (F \downarrow) \uparrow$.

С помощью псевдоинверсий можно сопоставить операции между сетевым исчислением min-plus и max-plus с помощью

$$(F \wedge G) \uparrow(n) = F \uparrow \vee G \uparrow \vee(n).$$

$$(F \otimes G) \uparrow(n) = F \uparrow \otimes G \uparrow(n).$$

$$(F + G) \uparrow(n) = \inf_{0 \leq K \leq n} \max\{F \uparrow(K), G \uparrow(n-K)\}.$$

Для отображения в другом направлении:

- $(F \vee G) \downarrow(t) = F \downarrow \wedge G \downarrow(t).$

- $(F \otimes G) \downarrow (t) = F \downarrow \otimes G \downarrow (t)$.
 - $(\inf_{0 \leq s \leq t} \max\{F(s), G(t-s)\}) \downarrow = F \downarrow (t) + G \downarrow (t)$.
- С этого, можно установить $\equiv T \downarrow A$ и $D \equiv T \downarrow D$, а также $\equiv TA \uparrow$ и $TD \equiv D' \uparrow$.
Кривые обслуживания и огибающие трафика связаны следующим образом:
- $D(t) \geq A \otimes S(t)$, $\forall t \longrightarrow TD(n) \leq TA \otimes s \uparrow (N)$, $\forall N$.
- $E(s) \geq a(t+s) - a(t)$, $\forall t, s \longrightarrow E \uparrow (M) \leq TA(N+M) - TA(n)$, $\forall n, M$.
 $TD(n) \leq TA \otimes GS(N)$, $\forall N \longrightarrow D(t) \geq A \otimes C \downarrow S(t)$, $\forall T$.
 $le(m) \leq TA(N+M) - TA(n)$, $\forall n, m \longrightarrow \lambda \downarrow E(s) \geq a(t+s) - a(t)$, $\forall t, S$.

16 Вероятностное сетевое исчисление

Сетевое исчисление применяет концепции кривых прибытия и кривых обслуживания, представленные ранее, для получения гарантий производительности как для отдельных, так и для объединенных сетевых элементов. Они включают в себя выходные данные, отставание и границы задержки. Две наиболее важные операции, применяемые сетевым исчислением, — это свертка с минимальным плюсом и свертка без свертки, как уже определено ранее. Вводится вероятностная структура и впоследствии результаты, полученные для детерминированного сетевого исчисления, немедленно распространяются на случай с несколькими узлами. Используется следующая нотация: Верхние индексы указывают узлы, соответствующие узлу, на который вводится поток, а нижние индексы указывают потоки. Определение кривых прибытия и обслуживания в определенном экземпляре u используются по местному времени.

Определение 4 (Местные кривые прибытия и обслуживания). Рассматривается возрастающая стохастическая последовательность $F(t)$, которая описывает кумулятивную функцию прибытия транспортного потока. Пусть $Au(t)$ — возрастающая стохастическая последовательность в t , такая, что для всех $t \geq 0$ и $u \geq t$

$$Au(t) \geq F(u) - F(u-t).$$

Тогда $Au(t)$ можно рассматривать как кривую прибытия $F(t)$ в локальном временном экземпляре u . Предполагается, что последовательность $F(t)$ является входом для сетевого элемента, а $F'(t)$ — соответствующим выходом. Пусть $Bu(t)$ — возрастающая стохастическая последовательность в t при $t \geq 0$, которая описывает доступную услугу в заданный момент времени $u \geq t$, такую, что

$$F'(t) \geq (F \otimes Bu)(t).$$

17 Циклический перебор дефицита (DRR)

Циклический перебор дефицита (DRR) — это широко распространенный алгоритм планирования, который обеспечивает справедливую очередь с пакетами переменной длины. Границы наихудших задержек для DRR были найдены Бойером и др., которые использовали строгий подход к се-

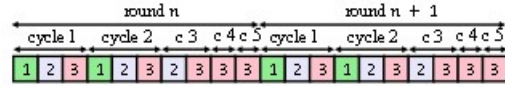
тевому исчислению и охарактеризовали обслуживание, полученное одним представляющим интерес потоком, с помощью выпуклой строгой кривой обслуживания. Эти границы не делают никаких предположений о мешающем трафике, следовательно, являются пессимистичными, когда мешающий трафик ограничен некоторыми кривыми прибытия. Для таких случаев были предложены два улучшения. В первом, написанном Сони и др., используется корректирующий термин, полученный из полужесткой эвристики; к сожалению, эти оценки неверны, как показывают в [16], демонстрируя контрпример. Последний, по Буйяру, строго выводит выпуклые строгие кривые обслуживания для DRR, которые учитывают ограничения кривой прибытия мешающего трафика. В статье [16] улучшают эти результаты двумя способами. Во-первых, получают невыпуклую строгую кривую обслуживания для DRR, которая улучшается по сравнению с Boyer et al. когда нет ограничений на прибытие мешающего трафика. Во-вторых, предоставляют итеративный метод для улучшения любой строгой кривой обслуживания (включая кривую Бульера), когда существуют ограничения на прибытие для мешающего трафика. На сегодняшний день результаты дают наиболее известные наихудшие оценки задержки для DRR. Они получены с помощью метода псевдоинверсии.

18 Взвешенный циклический перебор (WRR)

Взвешенный циклический перебор (WRR) часто используется, из-за его простоты, для планирования пакетов или задач. С помощью WRR количество пакетов, равного веса, выделенному потоку, может обслуживаться последовательно, что приводит к пакетному обслуживанию. Чередующийся взвешенный циклический перебор (WRR) — это вариант, который смягчает этот эффект. Авторы статьи [18] заинтересованы в нахождении границ задержки в наихудшем случае, полученной с помощью IWRR. С этой целью они используют подход сетевого исчисления и находят строгую кривую обслуживания для IWRR. Результат получается с использованием псевдоинверсии функции. Строгая кривая обслуживания является наилучшей из доступных и что границы задержки, полученные из нее, являются жесткими (т.е. наихудшими) для потоков пакетов постоянного размера. Кроме того, кривая строгого обслуживания WRR доминирует над кривой строгого обслуживания для WRR, которая была опубликована ранее. Приводится несколько численных примеров, иллюстрирующих сокращение задержек в наихудшем случае, вызванных IWRR, по сравнению с WRR. Взвешенный циклический перебор (WRR) — это алгоритм планирования, который часто используется для планирования задач или пакетов в системах реального времени или сетях связи. Емкость распределяется между несколькими клиентами или очередями, присваивая каждому из них вес, который является положительным целым числом, и предоставляя больше услуг тем, у кого больший вес. В частности, каждая очередь посещается одна за другой, и когда очередь i с весом w_i имеет возможность эмиссии, она отправляет пакеты

ты w_i или меньше, если присутствует меньше пакетов. Преимущество WRR заключается в том, что он справедлив и прост. Однако услуга является пакетной, поскольку для очереди i может последовательно обслуживаться до нескольких пакетов, что может привести к большому времени ожидания в худшем случае для других очередей. Чередующийся взвешенный циклический перебор (IWRR) смягчает этот эффект. С помощью IWRR очередь i с весом w_i имеет возможности выброса w_i за раунд и может отправлять до одного пакета при каждой возможности выброса. В отличие от WRR, он имеет одну возможность эмиссии за раунд и может отправлять до двух пакетов при каждой возможности эмиссии. Следовательно, IWRR распределяет возможности выбросов каждой очереди в раунде, что, как ожидается, приведет к более плавному обслуживанию и снижению задержек в худшем случае. Существует несколько версий IWRR; в данном случае сосредоточимся на простейшей, где очередь i имеет возможности для выбросов в первых циклах w_i в рамках раунда). Нас интересуют границы задержки для наихудшего случая, как это типично в контексте детерминированных сетей. С этой целью стандартным подходом является сетевое исчисление. В частности, с помощью сетевого исчисления услуга, предлагаемая интересующему потоку системой, абстрагируется с помощью кривой обслуживания. Оценка задержки в наихудшем случае получается путем объединения кривой обслуживания с кривой поступления интересующего потока. Кривая прибытия — это ограничение на объем данных, которые может отправить интересующий поток; такое ограничение необходимо для существования конечной границы задержки. Подход сетевого исчисления был применен к WRR в [17, раздел 8.2.4], где получена строгая кривая обслуживания. Кривая строгого обслуживания IWRR доминирует над кривой WRR, следовательно, результирующие границы задержки для IWRR всегда меньше или равны таковым для WRR. Строгая кривая обслуживания позволяет получить границы задержки с помощью сетевого исчисления, но такие границы не всегда могут быть жесткими, т.е. они не всегда могут быть равны наихудшим случаям. Это связано с тем, что строгая кривая обслуживания является абстракцией системы. Рассматривается взвешенная циклическая подсистема, которая обслуживает n входных потоков, имеет одну очередь на поток и использует взвешенный циклический алгоритм (описанный ниже) для арбитража между потоками. Взвешенная циклическая подсистема сама размещается в более крупной системе и может конкурировать с другими подсистемами массового обслуживания. Например, случай сервера с постоянной скоростью с несколькими уровнями приоритета, без вытеснения, и где взвешенная циклическая подсистема имеет уровень приоритета, который не является самым высоким. Предполагая некоторые ограничения кривой прибытия для трафика с более высоким приоритетом, услуга, получаемая всей взвешенной циклической подсистемой, может быть смоделирована с использованием строгой кривой обслуживания [17, раздел 8.3.2]. Это побуждает предположить, что совокупность всех потоков во взвешенной циклической подсистеме получает строгую кривую обслуживания, скажем, $\beta \in F$, которую называется «совокупной строгой кривой обслуживания».

Если взвешенная циклическая подсистема имеет эксклюзивный доступ к линии передачи со скоростью c , то $\beta(t) = ct$ для $t \geq 0$. Предполагается, что $\beta(t)$ конечно для каждого (конечного) t , и, не теряя общности, β является супераддитивным. Кроме того, нужно дополнительное техническое допущение, в первую очередь для установления результата герметичности: предполагается, что β является непрерывным по Липшицу, т.е. Существует константа $K > 0$, такая, что $\beta(t)\beta(s)t - s \leq K$ для всех $0 \leq s < t$; это не появляется быть ограничением, поскольку скорость, с которой передаются данные, имеет физический предел. Алгоритм арбитража, предполагаемый в этой статье [18], — это IWRR. Когда пакет потока i поступает в подсистему взвешенного циклического перебора, он помещается в очередь i . Вес потока i равен w_i . IWRR выполняет бесконечный цикл раундов. В одном раунде каждая очередь i имеет две возможности эмиссии; один пакет может быть отправлен в течение одной возможности эмиссии. Внутренний цикл определяет цикл, в котором посещается каждая очередь, но только те, у которых вес не меньше номера цикла, имеют возможность эмиссии. Предполагается, что команда отправки является единственной с ненулевой длительностью. Его фактическая продолжительность зависит от размера пакета, а также от объема услуг, доступных для всей взвешенной циклической подсистемы. Иллюстрацию смотрите на рисунке



Возможности выбросов в двух последовательных раундах для IWRR с тремя потоками и $w_1 = 2$, $w_2 = 3$, $w_3 = 5$. Это не временное поведение: каждая возможность может привести к пустому интервалу, если очередь пуста в это время. Кроме того, продолжительность каждого непустого интервала зависит от размера пакета и совокупного доступного сервиса (не предполагается обслуживание с постоянной скоростью).

19 Библиотека Nancy

В статье [19] описывается Nancy, библиотека сетевого исчисления (NC), которая позволяет пользователям эффективно выполнять сложные операции алгебры min-plus и max-plus. Насколько известно, Nancy — единственная библиотека с открытым исходным кодом, которая реализует операторы, работающие с произвольными кусочно-линейными функциями (в отличие от только вогнутых / выпуклых), а также для реализации некоторых из них (например, субаддитивное замыкание и композиция функций). Nancy позволяет исследователям вычислять результаты ЧПУ, используя простой синтаксис, который соответствует алгебраическому. Более того, он разработан с учетом вычислительной эффективности: он использует интеллектуальное представление данных, использует наследование для обеспечения более быстрых алгоритмов, когда они доступны (например, для опреде-

ленных подклассов функций), и он изначально параллелен, что позволяет использовать преимущества многоядерного оборудования. Это делает его пригодным для решения задач ЧПУ, которые ранее считались выходящими за рамки управляемых.

Функция в NS часто представляет объем данных, наблюдаемых в сетевом элементе до момента времени t . Соответственно, называют абсциссу - "временем", а его ординату - "данными". Библиотека Napsu предоставляет богатые классы, которые позволяют моделировать, манипулировать и выполнять вычисления с помощью кусочно-аффинных функций UPP. Свойство UPP гласит, что: $f(t + kd) = f(t) + k \cdot c \quad \forall t \geq T, \forall k \in \mathbb{N}$. Из вышесказанного следует, что для вычисления $f(t)$ для любого $t \geq 0$ нужно только сохранить его значения в $[0, T + d]$ и три параметра T , d и c . Это наглядно показано на рисунке 3. На левом графике изображена функция, переходная часть которой (до T) состоит из двух сегментов, а период (повторяющийся дважды) также состоит из двух сегментов.

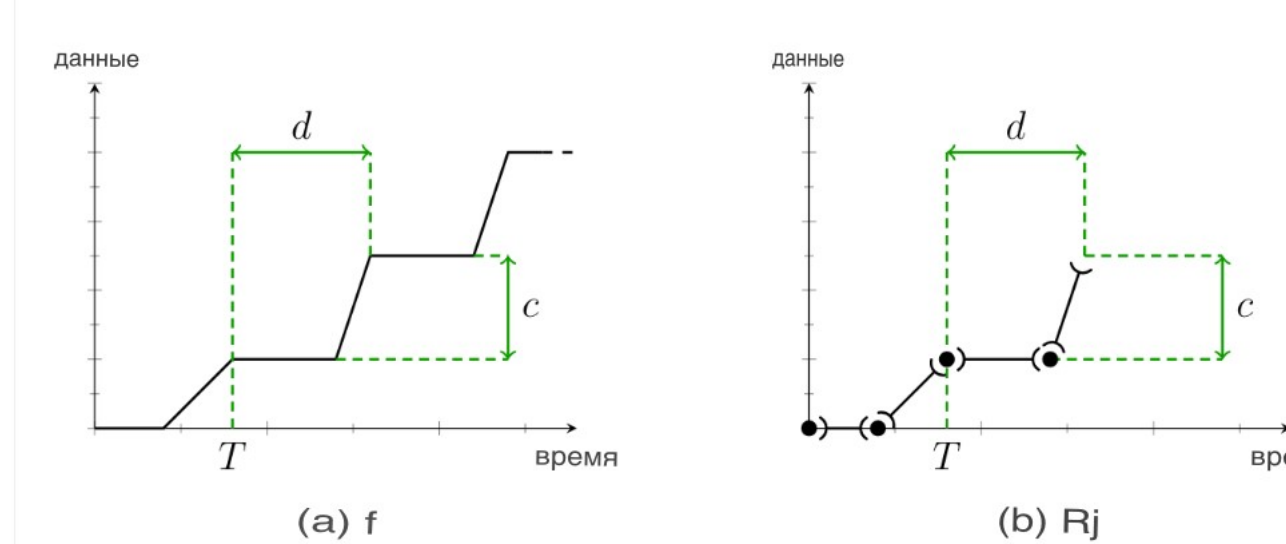


Рисунок 3: Пример в конечном счете псевдопериодической кусочно-аффинной функции f и ее представления Rf , взятый из [20].

В коде это представляется с помощью кривой класса, которая состоит из базовой последовательности и значений `PseudoPeriodStart` (T), `PseudoPeriodLength` (d) и `PseudoPeriodHeight` (c). Базовая последовательность имеет тип `Sequence`, который моделирует кусочно-аффинную функцию в ограниченной области, обычно замкнутую слева и открытую справа $[a, b]$. Последовательность состоит из элементов.

Элемент — это абстрактный класс, расширенный точкой (единственной парой значений времени) и сегментом (определенным в открытом интервале $[a, b]$). Это позволяет моделировать непрерывные кривые UPP. Правый график на рисунке 3 показывает переходную часть, состоящую из четы-

рех элементов, и a период, состоящий из четырех элементов. Четверка Rf , T , d , c - это информация, необходимая для представления вышеупомянутой функции в коде. Числовой тип, используемый во всей библиотеке, $Rational$, который представляет числа в $Q \cup \{+\infty, -\infty\}$. Это позволяет выполнять вычисления без каких-либо приближений (в отличие от того, что происходит при использовании двойного типа). Чтобы избежать проблем с переполнением, используется $BigInteger$, хотя мы поддерживается использование $long$ вместо этого для повышения производительности, используя флаг компиляции. Эти типы являются неизменяемыми, т.е. они не могут быть изменены после построения, и вместо этого методы будут создавать новые объекты для возврата. Это имеет двойное преимущество: с одной стороны, безопасность использования - объект никогда не изменит свойства, если он не будет переназначен. С другой стороны, это способствует распараллеливанию, поскольку все вычисления независимы и могут безопасно выполняться одновременно. Многие операторы \min - и \max -plus реализованы как методы этих классов и являются закрытыми, т.е. они возвращают объекты того же класса, к которому они принадлежат - фактически, они вычисляют ограниченное представление результата UPP. Это означает, что они могут быть связаны в цепочку, что позволяет вычислять сложные выражения NC. Основными реализованными операторами являются [21]:

- Минимум и максимум;
- Сложение и вычитание. Последний может вычислять как $A(t) - B(t)$, так и $[A(t) - B(t)]_+ = \max\{A(t)B(t), 0\}$ (по умолчанию);
- (минимум плюс) Свертка и деконволюция;
- Вертикальная и Горизонтальное отклонение;
- $\text{MaxPlusConvolution}$ и $\text{MaxPlusDeconvolution}$;
- Нижний и верхний параллелепипед;
- Суб- и супераддитивное раскрытие;
- Композиция, т.е. $h(t) = f(g(t))1$.

Также реализуется множество полезных методов и свойств для обычных манипуляций и проверок, таких как:

- Выборка кривой в момент времени или "рядом" с ValueAt , LeftLimitAt и RightLimitAt ;
- Свойства, такие как IsContinuous , IsNonDecreasing , IsConcave , IsSubAdditive и т.д.;
- Сдвиг вдоль оси времени (задержка, Precidateby) и ось данных (вертикальный сдвиг);
- Ограничение кривой конечным временным интервалом с помощью оператора Cut .

Как обсуждалось в [20], представление результата вышеуказанных операций часто намного больше, чем необходимо, т.е. оно включает в себя гораздо больше сегментов, чем требуется. Это может привести к серьезной неэффективности памяти / скорости, особенно при цепных операциях (напомним, что операторы ЧПУ часто используют алгоритмы, которые являются суперлинейными по количеству сегментов). Библиотека Nancy реализует эффективный алгоритм минимизации представления post-hoc , который вы-

полняется по умолчанию во всех операндах перед возвратом результата, что решает эту проблему. Кроме того, используется высокая распараллеливаемость вышеуказанных операторов с помощью PLINQ. Библиотека Nancy будет распараллеливать следующее:

- агрегированные операторы, такие как агрегированная свертка большого набора кривых;
- большие наборы независимых задач.

Например, свертка включает в себя вычисление нижней огибающей набора элементов во временной области. Элементы сначала группируются в (временные) сегменты в соответствии с их интервалом определения. Затем нижние огибающие для каждого сегмента вычисляются независимо параллельно. Наконец, многие методы класса Curve определены как виртуальные, так что его подклассы могут предоставлять более эффективные алгоритмы. Например, свертка вогнутых кривых заменяется их минимумом, который является менее сложным. Это делает библиотеку легко расширяемой для использования существующих или еще не найденных математических свойств. Например, были реализованы специализированные алгоритмы для SubAdditiveCurves, которые значительно более эффективны [20]. В качестве примера вышеизложенного в таблице 1 приводится время, необходимое для вычисления свертки между двумя субаддитивными кривыми, используя как стандартный алгоритм, так и оптимизированный в [20], как в последовательной, так и в параллельной настройке. Выполняются вышеуказанные вычисления на портативном компьютере (i710750H, 32 ГБ оперативной памяти). Таким образом, математические свойства, описанные в [20], сокращают время вычислений на порядки. Это подчеркивает, что гибкость системы типов, реализованной в Нанси, является большим преимуществом для будущих улучшений. С другой стороны, что распараллеливание приносит значительные преимущества даже для неоптимизированных вычислений. Это может быть обобщено на случаи, когда оптимизированные математические свойства недоступны или недостаточны (многие примеры которых можно найти в [20]). Однако попытка распараллелить оптимизированный алгоритм не дает никаких преимуществ: этого и следовало ожидать, поскольку время вычисления уже составляет менее 1 мс, поэтому накладные расходы, необходимые для распараллеливания, выше, чем сэкономленное время. Библиотека позволяет точно настраивать используемые алгоритмы с помощью параметра settings, который можно использовать, чтобы избежать неоптимальных настроек, таких как неправильное использование распараллеливания, как в приведенном выше случае.

	Standard	Optimized
Sequential	Q1: 20512.67 ms	Q1: 0.16 ms
	Q2: 20569.76 ms	Q2: 0.17 ms
	Q3: 20733.97 ms	Q3: 0.64 ms
Parallelized	Q1: 5860.67 ms	Q1: 0.49 ms
	Q2: 6030.53 ms	Q2: 0.52 ms
	Q3: 7657.40 ms	Q3: 2.22 ms

Таблица 1: Сравнение производительности $\beta 416, 835 + 313 \otimes \beta 552, 571 + 970$ с использованием распараллеливания и оптимизированных алгоритмов из [20]. Результаты представлены для каждого случая в виде 25-го перцентиля (Q1), медианы (Q2) и 75-го перцентиля (Q3) за 10 независимых прогонов.

20 Выводы

Теория Сетевого исчисления обеспечивает глубокое понимание проблем, возникающих в сети при обслуживании потоков трафика. Их использование позволяет понять некоторые фундаментальные свойства работы механизмов обслуживания мультисервисного трафика, таких, как архитектура управления ресурсами IntServ (интегрированное обслуживание), оконное управление потоком (window flow control), планировщики пакетов, формирователи трафика. Модульность и масштабируемость делает ее ценным инструментом анализа сложных систем и сетей. Сетевые модели могут быть широко использованы на всех отечественных при разработке как долгосрочных, так и текущих планов. Сетевое планирование позволяет не только определять потребность различных производственных ресурсов в будущем, но и координировать их рациональный расходв настоящем. С помощью сетевых графиков можно соединить в единую систему всематериальные, трудовые, финансовые и многие другие ресурсы и средства производства и в идеальных (планируемых), и в реальных (существующих) экономических условиях. Такие выводы приведены в статье [34]. Сетевое исчисление Network Calculus становится универсальной методологией анализа различных систем и сетей массового обслуживания. Перспективность данного подхода заключается в том, что он позволяет решать задачи, которые являются сложными для других методик, так как оперирует с граничными оценками исследуемых характеристик, а не стремится получить их точные значения. Высокая моделирующая способность сетевого исчисления совсем недавно была продемонстрирована на практике при решении традиционных для Internet инженерных сетевых задач, связанных с оценкой качества обслуживания QoS на основе технологий IntServ и DiffServ, а также при анализе различных сетей и систем массового обслуживания, таких как беспроводные сенсорные сети, коммутируемый Ethernet AFDX, системы-на-

кристалле, умные энергетические сети (smart grids), виртуальные частные сети VPN и др. [33]. Достигнутые успехи в теории НС позволяют прогнозировать, что пройдет определенное время и перечисленные выше нерешенные теоретические и практические задачи, и проблемы найдут свое эффективное разрешение.

21 Список используемых источников

1. Centralized QoS Routing Using Network Calculus for SDN-Based Streaming Media Networks SHENG ZHU^{1,2}, ZHEN SUN^{1,2}, YONG LU^{1,2}, LIANMING ZHANG^{1,2}, YEHUA WEI^{1,2}, AND GEYONG MIN³, (Senior Member, IEEE)
2. БАЗОВЫЕ ПРИНЦИПЫ И ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ТЕОРИИ СЕТЕВОГО ИСЧИСЛЕНИЯ (NETWORK CALCULUS) Кудрявцева Е.Н., Росляков А.В.
3. An End-to-End Probabilistic Network Calculus with Moment Generating Functions for Efficient Utilization of Independence Markus Fidler[†] Centre for Quantifiable Quality of Service (Q2S)[‡] NTNU Trondheim, Norway
4. ESTIMATING TIME CHARACTERISTICS OF CONTROL SYSTEMS WITH CYCLIC OPERATION: A NETWORK CALCULUS APPROACH V.G. Promyslov² and K.V. Semenov³ Trapeznikov
5. Tight performance bounds in the worst-case analysis of feed-forward networks Anne Bouillard, Eric Thierry
6. Le Boudec J.-Y., Thiran P. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, Springer, LNCS, 2001.
7. Jiang Y, Liu Y Stochastic Network Calculus. London: SpringerVerlag, 2008.
8. Network Calculus. Theory and application A.V. Roslyakov Chair of Automatic Telecommunications, Povolzhskiy State
9. Le Boudec J.-Y., Thiran P. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet. – Springer, LNCS, 2001.
10. Extending the Network Calculus Algorithmic Toolbox for Ultimately Pseudo-Periodic Functions: Pseudo-Inverse and Composition Raffaele Zippo^{1,2,3*}, Paul Nikolaus³ and Giovanni Stea²
11. Network Calculus with Flow Prolongation A Feedforward FIFO Analysis enabled by ML Fabien Geyer Alexander Scheffler Steffen Bondorf
12. Sensor Network Calculus with Multiple Sinks Jens B. Schmitt¹, Frank A. Zdarsky¹, Utz Roedig²
13. ITU-T Recommendation Y.1315. QoS support for VPN services – Framework and characteristics. [Электронный ресурс] – Режим доступа: <https://www.itu.int/rec/T-REC-Y.1315200609-I/en.html>
14. Росляков А.В., Лысиков А.А. Применение теории стохастических сетевых исчислений к анализу характеристик VPN // T-Comm. Телекоммуникации и транспорт. – 2013. – №7. – С.106–108.
15. A Fluid-Flow Interpretation of SCED Scheduling J. org Liebeherr Department of Electrical and Computer Engineering University of Toronto
16. Deficit Round-Robin: A Second Network Calculus Analysis Seyed Mohammadhossein

- Tabatabaee, Member, IEEE, and, Jean-Yves Le Boudec, Fellow, IEEE
17. A. Bouillard, M. Boyer, and E. Le Corronc, Deterministic Network Calculus: From Theory to Practical Implementation. Wiley-ISTE
 18. Interleaved Weighted Round-Robin: A Network Calculus Analysis Seyed Mohammadhossein Tabatabaee, Jean-Yves Le Boudec Marc Boyer
 19. Nancy: an efficient parallel Network Calculus library Raffaele Zippo, Giovanni Stea Dipartimento di Ingegneria dell'Informazione, University of Pisa, Italy. Dipartimento di Ingegneria dell'Informazione, University of Florence, Italy.
 20. R. Zippo, G. Stea, Computationally efficient worst-case analysis of flow-controlled networks with network calculus
 21. A. Bouillard, M. Boyer, E. Le Corronc, Deterministic Network Calculus: From Theory to Practical Implementation, Wiley, 2018.
 22. On the Robustness of Deep Learning-predicted Contention Models for Network Calculus Fabien Geye, Steffen Bondorf
 23. Study on MCS Selection and Spectrum Allocation for URLLC Traffic under Delay and Reliability Constraint in 5G Network Yuehong Gao, Changhao Sun, Xiaonan Zhang and Xiao Hong
 24. J. B. Schmitt, F. A. Zdarsky, and M. Fidler. Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch ... In Proc. of INFOCOM'2008, 2008.
 25. О ТЕОРИИ ОЧЕРЕДЕЙ И СЕТЕВОМ ИСЧИСЛЕНИИ ДЛЯ ОБЪЕКТОВ ПОВЫШЕННОЙ ОПАСНОСТИ Байбулатов А.А.
 26. Baybulatov A.A. A Novel Approach to Estimating Databases Maximum Updating Time // Advances in Intelligent Systems and Computing. – 2018. – Vol. 658. P. 104-112.
 27. A Stochastic Network Calculus Yong Liu a, Chen-Khong Tham a, Yuming Jiang b
 28. Comparison of different classes of service curves in Network Calculus Anne Bouillard Laurent Jouhet Eric Thierry
 29. Andreozzi, M., Conboy, F., Stea, G., Zippo, R.: Heterogeneous systems modelling with adaptive traffic profiles and its application to worst-case analysis of a dram controller. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 79–86 (2020). IEEE
 30. Network calculus for parallel processing G. Kesidis, Y. Shan, B. Urgaonkar J. Liebeherr
 31. Of Kernels and Queues: when network calculus meets analytic combinatorics Anne Bouillard¹, C eline Comte^{1,2}, Elie de Panafieu¹, and Fabien Mathieu¹
 32. Trade-off between accuracy and tractability of network calculus in FIFO networks Anne Bouillard
 33. Lysikov A.A., Roslyakov A.V. Teoriya Network Calculus i yeye primeneniye k issledovaniyu sistem avtomatiki i elektroniki
 34. СЕТЕВОЕ ИСЧИСЛЕНИЕ NETWORK CALCULUS. ЧАСТЬ 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ Росляков А.В., Лысиков А.А., Витевский В.Д.
 35. Unleashing the Power of Paying Multiplexing Only Once in Stochastic Network Calculus Anne Bouillard, Paul Nikolaus, Jens Schmitt
 36. A Basic Stochastic Network Calculus Yuming Jiang Department of Telematics

Norwegian University of Science and Technology