МОДЕЛЬ ДОСТУПНОСТИ НА ОСНОВЕ ТЕОРИИ СЕТЕВОГО ИСЧИСЛЕНИЯ ДЛЯ ПОТОКОВОЙ СИСТЕМЫ ОБРАБОТКИ ДАННЫХ[1]

Промыслов В. Г.[2],
(*ФГБУН Институт проблем управления им. В.А. Трапезникова РАН, Москва*)

Анализируется проблема проектирования системы с учетом доступности для цифровых вычислительных систем, ориентированных на потоковую обработку данных. Доступность рассматривается в контексте модели «конфиденциальность, целостность доступность (КЦД)» информационной безопасности. Доступность характеризуется тем, что она является оценкой временных свойств системы, ее функции или компонента в заданных условиях в конкретный момент времени. Для оценки доступности предложена трехуровневая референтная модель, связанная с различным представлением системы на архитектурном и функциональном уровне. Рассмотрена реализация модели в рамках теории сетевых исчислений (ТСИ), что позволяет рассчитать предельные (консервативные) оценки временных параметров в системе. Показана применимость модели с ТСИ в инженерной практике для анализа доступности в распределенных цифровых вычислительных системах с конкурентной обработкой потоковых данных. Полученная оценка доступности может быть использована для диагностики отклонений поведения системы в результате ошибок или враждебных действий злоумышленника, а также при проектировании для обоснования архитектуры и характеристик компонентов системы.

Ключевые слова: доступность, модель, проектирование, теория сетевых исчислений, анализ, информационная безопасность.

1. Введение

Современные цифровые системы управления, особенно для технических объектов, имеют достаточно сложную архитектуру, выражающуюся в наличии распределенной структуры, в рамках которой компоненты обмениваются информацией для выполнения одной из функций системы. Необходимость информационного взаимодействия в системе накладывает ограничения на характе-

[1] *Исследование выполнено за счет гранта Российского научного фонда № 23-19-00338, https://rscf.ru/project/23-19-00338.*
[2] *Виталий Георгиевич Промыслов, к.ф.-м.н., в.н.с.* (*vp@ipu.ru*).

ристики компонентов и потоки информации в ней. Такие ограничения учитываются при проектировании и разработке системы. В современной практике, в соответствии с принципом безопасного проектирования [30], важным типом ограничений являются ограничения, связанные с информационной безопасностью. В информационной безопасности (ИБ) принято характеризовать качество информации через такие свойства как конфиденциальность, целостность и доступность, и совокупность трёх базовых свойств ИБ называется моделью КЦД. Модель была предложена Зальцером и Шрёдером (Saltzer и Schroeder) в семидесятых годах XX века [35]. Выбор трех свойств связан с принятой концепцией ИБ, которая позволяет описать любое событие ИБ через три свойства КЦД. Данная классификация свойств безопасности сохранилась в большинстве современных моделей информационной безопасности [31]. Однако если сам набор свойств в основном стабилен, то в последнее время в связи с усилением роли информационной безопасности в промышленных системах, связанных с реальным временем, происходит перераспределение их приоритета. В частности, на первый план выступают свойства, связанные с доступностью [9]. Термин «доступность» используется во многих дисциплинах, связанных с техническими системами, и его интерпретация зависит от того, в рамках какой дисциплины рассматривается доступность. Например, в контексте классической надежности обычно в качестве параметров доступности рассматривают такие усредненные параметры как наработка на отказ, время готовности и др. [8]. Для оценки временных параметров систем в рамках задач надежности разработаны многочисленные методы, из которых стоит выделить методы теории массового обслуживания (ТМО). В применении ТМО для компьютерных и цифровых систем имеется многочисленная литература, например, ставшая классической работа [4]. Однако использование ТМО для оценки доступности в контексте информационной безопасности не всегда оправдано; это связано с одной стороны, со значительными трудностями расчета систем в случае нетривиальных статистических распределений для потоков и времени обслуживания в системе [3, 5], с другой стороны – с особенностями интерпретации доступности в информационной безопасности: так, в работе [21]

выделяют шесть видов интерпретации доступности, которые связаны как с внешними, так и внутренними характеристиками и состоянием системы. В основном свойство доступности связано с временными параметрами системы, причем используются как мгновенные оценки доступности, так и усредненные, вероятностные оценки [21]. Однако в ИБ, в отличие от других дисциплин, чаще используют именно мгновенные оценки. Например, определение, данное в серии стандартов МЭК 62443 [9] и принятое в работе, характеризует доступность как «способность компонента выполнить требуемое действие при заданных условиях в заданный момент времени или в продолжение заданного интервала времени, если предоставлены необходимые внешние ресурсы». Определение подчеркивает неразрывную связь доступности с временными характеристиками и важность гарантированности отклика на заданное воздействие в определенном интервале времени. Необходимо отметить, что доступность в ИБ не ограничивается временными характеристиками, хотя они являются основными, она может зависеть от показателей надежности, удобства сопровождения и качества технической поддержки и наличия прав доступа, однако данные характеристики не рассматриваются в работе. Для их оценки в вычислительных сетях можно применять хорошо разработанные риск-ориентированные подходы [10].

Для временных характеристик доступности встает вопрос о выборе «подходящего» метода проектирования и анализа систем для оценки требуемых параметров доступности. Для задач безопасности, в частности, были предложены референтная модель и метрика, которые декомпозируют свойство доступности в виде ограничений на суммарную задержку передачи информации между узлами системы [2, 15], и для расчета ограничений предполагалось применить теорию сетевых исчислений [16]. Термин «референтная модель» используется в смысле обобщенного описания системы с указанием основных ее составляющих и связей между ними. Теория сетевых исчислений (ТСИ) (Network Calculus) – это подход [22, 23] к анализу потоковых систем, который позволяет описать архитектуру системы в виде набора ограничений на основные параметры, такие как скорость

и неравномерность потоков, производительность узлов и получить оценки для параметров, связанных с временем и буферизацией данных, также в виде ограничений. В основном ТСИ нашло применение для анализа цифровых систем передачи и обработки данных реального времени. В работе [16] применение ТСИ для оценки доступности цифровой системы управления иллюстрируется на примере тривиальной системы с последовательной обработкой данных и простой топологией без ветвлений и циклов порождающих взаимовлияний потоков на компонентах.

В работах [2, 15] модель доступности выражена как композиция элементов, включающая метрики доступности, множество функций, архитектуру системы, описание используемой платформы, множество задержек и временных параметров. В качестве меры доступности используется время передачи сигнала от источника до приемника, которое сравнивается с заданным в спецификации временем на выполнение функции в системе с последовательным соединением компонентов и единственным потоком на входе.

Очевидно, что подход к оценке доступности, методика ее расчета зависят как от топологии исследуемой системы, так и свойств конкретных компонентов и внешних условий, выраженных во входных потоках информации, поступающих в систему.
Цель данной работы – развить референтную модель доступности [15, 16] для задач проектирования и анализа цифровых распределенных систем общего вида.

Поэтому, в отличие от ранее сформулированной дополненной референтной модели, основное внимание будет уделено двум аспектам:

- расширению применимости модели доступности на распределенные цифровые системы с конкурентными потоками данных;
- разработке «вычислимой» модели, приспособленной для инженерных расчетов доступности вычислительных систем, для которых граф, описывающий информационную структуру, не содержит циклов.

В частности, будут рассмотрены системы, не требующие топологии сети в виде дерева, но предполагающие «прямую» обработку информации в компонентах (узлах) («feed forward») системы [36].

Научная новизна работы заключается в том, что в дополненной модели доступность может оцениваться на каждом из трех уровней представления системы: архитектурном, платформы и функциональном. Для этого предлагается применить ТСИ не только для оценки граничных значений отдельных параметров в узлах системы, но и для расчета обобщенных детерминированных характеристик, позволяющих рассчитать временные характеристики многокомпонентных систем с конкурентными потоками на различных уровнях их представления. Для систем с прямой обработкой данных в рамках развития референтной модели доступности будет дана ее практическая интерпретация в терминах ТСИ.

В работе также приведены и дополнены некоторые основные положения ТСИ, необходимые для понимания интерпретации модели доступности, однако для более полного ознакомления с теорией следует обратиться к специальной литературе, например [32].

Системы с циклической зависимостью алгоритмов обработки информации между компонентами системы не обсуждаются в работе, расчет таких систем возможен в ТСИ для некоторого класса систем, методика расчета предложена в [23] и развита далее многими авторами [28]. В отличие от систем без циклической зависимости, расчет налагает серьёзные ограничения на параметры системы, связанные со сходимостью используемых алгоритмов расчета [32]. Более подробно применения ТСИ для систем, содержащих циклические зависимости, исследуется в работах [36, 37, Гл. 6].

Заметим, что в рамках расширений ТСИ возможно соотнести параметры, полученные посредством ТСИ, со статистическими оценками для ТМО [20, 29]. Данные подходы расширяют основные понятия ТСИ для случая вероятностного подхода к основным определениям ТСИ, что приводит к возможности перехода от ТСИ к ТМО и обратно.

Однако объединение ТСИ и ТМО вместе с преимуществом комплексного подхода к оценке временных параметров системы вводит в ТСИ проблему интерпретации величин, имеющих нетривиальное распределение, и методов работы с ними, что с инженерной точки зрения снижает прозрачность получаемых результатов. Вместе с тем простота и прослеживаемость вычислений, определяющие возможность проведения исчерпывающего анализа и тестирования системы, являются важным свойством для задач информационной безопасности, во многом связанных с ее культурой доверия [7]. Поэтому далее ТСИ будет рассматриваться в ее «классическом» детерминированном виде.

В работе приведен пример расчёта доступности для модели системы с применением ТСИ, а также показана возможность реализации системы в виде макета с параметрами ограниченными ТСИ. Основной задачей как моделирования, так и проверки на экспериментальном макете является показать преимущества применения ТСИ в качестве метода оценки доступности.

2. Модель ТСИ

2.1. ОСНОВНЫЕ ЭЛЕМЕНТЫ ТСИ

Для оценки доступности предлагается использовать метод Network Calculus [22, 23], также известный как теория сетевых исчислений (ТСИ). В своем классическом варианте ТСИ описывает систему в виде набора ограничений на потоки информации, передаваемые в системе, а также на ресурсы, которые каждый компонент может представить для потока.

Рассмотрим систему S, представленную на рис. 1.

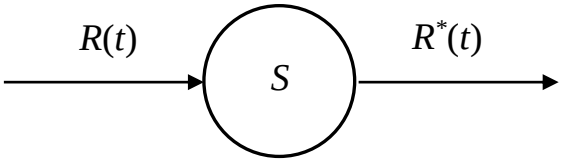


Рис. 1. Простая модель системы в контексте ТСИ

Система S рассматривается как чёрный ящик, который на вход получает входной поток данных r, а на выходе выдаёт выходной поток r^*. В ТСИ обычно рассматривается не мгновенная реализация потока r, а его представление в виде неубывающей функции − кумулятивного потока R.

Определение. (Кумулятивная) функция потока есть неотрицательная неубывающая функция времени:

$R(t) = 0, T < 0; R: \mathbb{R} \to \mathbb{R}_+ \cup \{+\infty\}, \mathrm{R}(t) \leq R(s), \forall t < s.$

Функция потока может рассматриваться как интеграл

(1) $R(t) = \int_0^t r(u)\mathrm{d}u$, при $r(t) \geq 0, t \geq 0.$

Считаем, что в системе предполагается отсутствие потерь:

(2) $R(t) = R^*(t), t \to \infty,$

и добавление информации при ее обработке на серверах:

(3) $R(t) \geq R^*(t), \forall t \geq 0.$

При таких ограничениях выполняется принцип казуальности (причинности).

Определение. Функция потока является казуальной, если $R(t) = 0 \quad \forall t < 0.$

Заметим, что для систем с потерями есть расширения ТСИ [24], однако в настоящей работе система считается казуальной, т.е. для каждого компонента выполняются условия (2) и (3). Такое допущение не влияет существенно на применимость ТСИ в референтной модели доступности для систем потоковой обработки и может означать нулевые начальные условия в момент запуска системы.

Для описания системы в ТСИ используются операторы мини(макси)-плюс алгебры [37]. Будем использовать нотацию, введённую в рамках работы [32]. В частности, будут использоваться обозначения $\otimes$ для операции миниплюс конволюции и $\oslash$ – для миниплюс деконволюции.

В ТСИ для описания классов потоков и ресурса, доступного на серверах при их обработке, используются их представления в виде ограничений: огибающей (конверта) потока и кривой обслуживания. Переход от потока и ресурса к их ограничениям позволяет не рассматривать конкретную реализацию потока, а работать с классом потоков, удовлетворяющих ограничениям.

Определение. Огибающей потока $R(t)$ называется такая казуальная функция $\alpha(t)$, что для $\forall s, t \geq 0$, $s \leq t$:

(4) $R(t) - R(s) \leq \alpha(t-s)$.

Огибающая потока может для быть выражена через оператор миниплюс обратной свертки:

(5) $\alpha \geq R \oslash R$.

Легко видеть, что $\forall T > 0$ $\alpha(T)$ является наибольшим объемом данных, поступившим с потоком $R(t)$ за время T. Практические рекомендации по расчету огибающей потока можно найти в работе [1].

Кривая обслуживания для одного потока в ТСИ определяет ограничения на ресурс, который занимает сервер на обслуживание каждого потока данных. Данная характеристика является имманентной, не зависящей от внешних условий характеристикой системы. Для определения характеристик системы ТСИ в рамках выходных потоков введем определение строгой кривой обслуживания системы [32].

Определение. Функция β – это (минимальная) кривая обслуживания системы S с входным потоком R и выходным потоком R^*:

$R^*(t) - R(s) \geq \beta(t-s)$ $\forall s, t \geq 0, s \leq t$.

Кривая обслуживания может быть выражена через оператор миниплюс свертки:

(6) $R^* \geq R \otimes \beta$.

Практические рекомендации по расчету кривой обслуживания можно найти в работе [13].

В ТСИ важное значение имеют кривые обслуживания специального вида – строгие кривые обслуживания (strict service curve).

Определение. Функция β является строгой кривой обслуживания системы S с входным потоком R и выходным потоком , если для любого периода времени $\forall t, s \geq 0, s \leq t, u = t - s$:

$R^*(u) \geq \beta(u)$.

Результатами расчета в ТСИ являются ограничения (наихудшие и наилучшие) на две основные характеристики системы *S*:

задержку при обработке данных в системе;

размер буферизируемых данных.

Для оценки доступности наибольшую важность имеет максимальная задержка обработки данных в системе D_{max} [32]:

(7) $D_{max} = h(\alpha, \beta)$,

где α и β – огибающие потока и кривая обслуживания. Она задаётся следующим выражением [5]:

(8) $h(\alpha, \beta) = \inf\{d \geq 0 : (\alpha \oslash \beta)(-d) \leq 0\}$
$= \sup\limits_{t \geq 0}\{inf\{d \geq 0 : \alpha(t) \leq \beta(t + d)\}\}$.

2.2. ОПИСАНИЕ СИСТЕМЫ В ТСИ

В предыдущем параграфе система, описываемая ТСИ, была тривиальной, реальные системы имеют более сложную структуру. В частности, на одном сервере могут обрабатываться несколько потоков, несколько серверов могут быть расположены на едином ресурсе и т.д.

Рассмотрим оценку доступности для цифровых систем с потоковой обработкой данных. Проблемным вопросом при анализе систем с применением ТСИ является выделение характеристик, связанных с конкретным потоком и/или конкретным компонентом из интегральных характеристик системы, так как совместная обработка потоков, их агрегирование на сервере в общем случае не являются линейной операцией.

Необходимо также учитывать, что результаты анализа агрегирования потоков зависят от дисциплины обработки (планировщика) отдельных потоков на едином ресурсе, который описывается кривой обслуживания. Далее мы рассмотрим только две основные дисциплины: последовательная обработка (англ. FIFO) и дисциплина с произвольным выбором (Arbitrary). Расширение ТСИ для других дисциплин можно найти в работе [19].

Основой для анализа нетривиальных систем в ТСИ является набор теорем, представленных в работах [17, 23, 32]. Они позволяют получить ограничения на основные характеристики одного потока при условии наличия другого потока. Формулировки теорем приведем ниже.

Теорема 1. (О последовательном объединении серверов). *Пусть в системе S есть два последовательно соединённых сервера с кривыми обслуживания β^1, β^2. Тогда общая кривая обслуживания для системы может быть выражена как*

(9) $\beta = \beta^1 \otimes \beta^2$.

Доказательство см. [37 с. 28].

Теорема 2. (Огибающая выходного потока). *Пусть поток f ограничен кривой α и обрабатывается в сервере s с кривой обслуживания β. Тогда выходной поток f' будет ограничен огибающей α':*

(10) $\alpha'(t) = \begin{cases} 0, t \le 0, \\ \alpha \oslash \beta(t). \end{cases}$

Доказательство см. [37, с. 7].

Определение. Будем называть кривую обслуживания остаточной кривой обслуживания β^{lo}, если она является кривой обслуживания, которая может быть выделена компонентом для обработки одному потоку при условии, что на нем уже обрабатывается другой более высокоприоритетный поток.

Теорема 3. *Рассмотрим систему S, которая на сервере s имеет строгую кривую обслуживания β и обрабатывает два потока f_1 и f_2 с огибающими α_1 и α_2. При произвольной дисциплине обслуживания потоков на компоненте (arbitrary/blind multiplexing), когда порядок обслуживания потоков не определен, поток f_1 получит остаточную кривую обслуживания не хуже, чем*

(11) $\beta^{lo} = \sup_{0 \le s \le t}(\beta(s) - \alpha_2(s))$.

Доказательство см. [37, с. 176]. Заметим, что в [32] приведена несколько другая запись, связанная с тем, что в формулировке опущен оператор sup. В этом случае необходима специальная оговорка о казуальности $\beta_1(t)$.

Теорема 4. *Рассмотрим систему S, которая на сервере s имеет кривую обслуживания β и обрабатывает два потока f_1 и f_2 с огибающими α_1 и α_2. При последовательной дисциплине обслуживания поток f_1 получит остаточную кривую обслуживания не хуже, чем*

(12) $\beta_\theta^{lo} = \sup_{0 \le s \le t}(\beta(s) - \alpha_2(s - \theta))1(t > \theta)$,

где $\theta \ge 0$ – некоторый произвольный параметр, β_θ^{lo} описывает семейство кривых, которые, если удовлетворяют условию казуальности, также являются кривыми обслуживания для потока x_1.

Доказательство см. [37, с. 177]

Рассмотрим следствие теоремы 4 в виде модификации дисциплины последовательного обслуживания с ожиданием. Такая

дисциплина может быть реализована в компонентах с циклическим алгоритмом обработки поступающей информации на основе опроса наличия информации во входном канале без блокирования. Опрос происходит с некоторым периодом T; если информации нет, то программа засыпает на время T, а если информация есть, то она обрабатывается в порядке поступления. Однако если время обработки меньше, чем период опроса, то система блокируется на оставшееся от этого время.

Одной из кривых обслуживания для такой системы является кривая вида $\beta(t-T)$, где $\beta(t)$ – кривая обслуживания для системы без блокирования. Данное выражение легко выводится, если представить систему в виде последовательного соединенных компонента с фиксированной задержкой и компонента обработки потока. Однако такая оценка будет достаточно грубой. Лучшая оценка может быть получена, если учесть физический смысл параметра θ в формуле (12) как наименьшего времени, когда, во-первых, компонент отработал всплеск данных потока f_2 и, во-вторых, установившаяся скорость поступления данных для f_2 меньше, чем производительность сервера. В этом случае можно считать, что при $t \geq \theta$ у сервера есть ресурсы для обработки порции данных потока f_1.

Следствие 1. Рассмотрим систему *S*, которая реализована на компоненте u, имеет кривую обслуживания β и обрабатывает два потока f_1 и f_2 с огибающими α_1 и α_2. Обслуживание происходит не чаще, чем за интервал T. Тогда при последовательной дисциплине обслуживания на компоненте поток x_1 получит кривую обслуживания не хуже, чем:

(13) $\beta_\theta^{lo}(t) = \sup_{0\leq s\leq t}(\beta\big(s-(T-\theta)^+\big) - \alpha_2(s-\theta)\big)1(t>\theta)$,

где $\theta \geq 0$ – некоторый произвольный параметр, надстрочный символ + обозначает функцию вида

$$(T-\theta)^+ = \begin{cases}(T-\theta), \theta < T,\\ 0, T \leq \theta;\end{cases}$$

(t) – булева функция:

$$1(x) = \begin{cases}1, x == true,\\ 0, x == false;\end{cases}$$

β_θ^{lo} описывает семейство кривых, которые, если они удовлетворяют условию казуальности, также являются кривыми обслуживания для потока f_1.

Доказательство следствия следует из анализа функционирования системы. Для систем, где θ превышает T, кривая обеспечивает реализацию дисциплины последовательного обслуживания. Если θ меньше T, то кривая обслуживания всей системы представляет собой кривую обслуживания с последовательным компонентом задержки на $(T-\theta)$, ч.т.д.

Для удобства работы с формулами (11)–(13) введем оператор $\ominus$:

(14) $\beta \ominus (\theta,\alpha) = \sup_{0\le s\le t}(\beta(s-(T-\theta)^+ - \alpha(s-\theta))1(t>\theta)$.

Теоремы 1-4 и следствие 1 совместно с основными положениями ТСИ (п. 2) об ограничениях на задержку в системе позволят рассчитать для систем, где компоненты удовлетворяют условиям теорем 2-4, ограничения на задержку как между серверами где происходит обработка данных, так и интегральную задержку для потока на его пути.

Как указано выше, анализ системы в рамках ТСИ сталкивается с трудностями описания взаимодействия потоков, если топология системы содержит циклы. Далее мы сосредоточимся на классе систем, относящихся к системам без циклических зависимостей в путях обработки потоков (feed forward).

Рассмотрим референтную модель описания доступности в системе, построенную в рамках ТСИ, для чего сначала приведем основные положения модели и покажем ее применение в оценке доступности.

3. Референтная модель доступности

В работах [2, 15] предложена референтная модель доступности, которая выделяет шесть основных уровней в задаче оценки доступности: метрика, функция, система, платформа, задержка, временные параметры. Основываясь на данной модели, предложим модель, более ориентированную на практическое применение, где разделены элементы, отражающие абстрактное представление системы, и, собственно, методы оценки доступности,

а также расширена применимость модели для систем со многими потоками.

Модифицированная модель содержит пять элементов: три основных элемента определяют уровни представления системы, используемые для анализа, и два вспомогательных элемента, связанных с методом расчета доступности. Основные элементы:

- Архитектурный уровень, который задается множеством компонентов платформы и отношениями между ними. Примером такого описания может быть модель информационных потоков в исследуемой системе.
- Уровень платформы, который задается множеством компонентов системы и их показателями, например, задержки в максимальном времени обработки данных на компоненте.
- Функциональный уровень, который задается совокупностью действий анализируемой системы, направленной на достижение определенной цели, которая выражается в терминах архитектурного уровня или платформы. Например, характеристика «время архивирования данных» в системе складывается из времени прохождения данных по тракту обработки плюс время записи данных на носитель на сервере, выполняющем архивирование.

Вспомогательные:

- Метрика доступности A, позволяет задать множество времен доступа к информации $D = \{d_i\}$ (задержку) для каждого множества элементов анализируемого уровня системы.
- Барьерная функция L.

Определение. Метрика доступности A – это функция, которая каждому элементу основного уровня представления системы ставит в соответствие некоторое вещественное число, называемое задержкой: $A: M \to D, D \subseteq \mathbb{R}$. Далее, когда говорится о метрике, под ней будет пониматься метрика доступности.

Определение. Барьерной функцией L для оценки доступности E называется непрерывная функция, определенная внутри допустимого множества задержек D, ставящая для каждой задержки $d \in D$, рассчитанной по метрике A, значение {true, false}, в зависимости от того удовлетворяет задержка заданным условиям или нет, $L: D \to \{true, false\}$.

Для того чтобы определить, что такое доступность, положим, что оценка доступности E разбивается на серию индивидуальных задач оценки доступности $e \in E$, где каждая отдельная задача e может быть разрешена при помощи некоторого единого эффективного условия $l \in L$:

$$l(d) = \begin{cases} 1, 0 \le d \le d_{max}, \\ 0, d > d_{max}. \end{cases}$$

Определение. Оценкой доступности E, или просто доступностью, назовем операцию логического ИЛИ для значений e по совокупности индивидуальных задач:

(15) $E: L \times D \to \{true, false\}$,

(16) $E(D) = \bigvee_{i \in N, d_i \in D, d_{i\,max} \in D} l\left(d_i\ , d_{i\,max}\right)$.

Для инженерных задач является обычным представление системы в виде графа, где узлы графа представляют собой компоненты, где происходит обработка информации, а ребра графа представляют собой отношения, возникающие между узлами при выполнении определенной функции системы. В работе для простоты предполагается, что все узлы – это компоненты, обрабатывающие по какому-либо алгоритму информацию, а ребра – это линии связи между компонентами, по которым могут идти информационные потоки. Каждый из уровней представления системы является обобщением представления системы на более низком уровне.

Рассмотрим подробнее основные уровни, а также предложим для каждого метрику и барьерную функцию.

3.1. АРХИТЕКТУРНЫЙ УРОВЕНЬ

Рассмотрим архитектурный уровень Σ, который представлен потоками информации, циркулирующими в системе. Система состоит из серверов обрабатывающих потоки данных, соединённых направленными связями.

Определим основные элементы архитектуры, посредством которых можно рассчитать доступность потоков данных. Пусть:

$S = \{s_i\}$ – множество серверов в системе;

$F = \{f_i\}$– потоки информации. Поток задается в виде пути в графе G от s_iк s_j $s_i, s_j \in S$ и огибающей α **Ошибка! Источник ссылки не найден.**) на входе.

Каждый сервер описывается кортежем (F, l, β), где

$\beta: F \to F$ – передаточная функция сервера, заданная в виде кривой обслуживания β, **Ошибка! Источник ссылки не найден.**, и дисциплины обслуживания. Считаем, что преобразование является монотонным: $\forall f', f'' \in F\ f' \geq f'' => \beta(f') \geq \beta(f'')$.

$l: F \to \{true, false\}$ – барьерная функция проверки соответствия входных потоков условиям. Функция $l(F) = true$, если при заданных параметрах потока сервер может выполнить преобразование β за указанное время.

Тогда $G = (S, F)$ – конечный помеченный ориентированный граф, представляющий текущие доступы в системе. Элементы множеств *S* являются вершинами графа. Элементы множества *F* являются ребрами графа, $|F| = |S| \times |S|$.

На графе G определено множество маршрутов $\mathcal{P}, |\mathcal{P}| = |F|$, т.е. для каждого из потоков f_i в графе G определен маршрут из набора последовательных ребер, соединяющих начало и конец маршрута $\mathcal{P}_i$, через вершины графа, где происходит обработка потока:

(17) $\mathcal{P}_i = \cup_{k=1}^{l_j} (s_{i,k-1}, s_{i,k}), (s_{i,k-1}, s_{i,k}) \in E.$

Метрика задаётся отображением

(18) $A_\Sigma: F \to D.$

Метрика сопоставляет для каждого потока информации некоторую величину, связанную с задержкой передачи информации по маршруту, связанному с потоком.

Оценка доступности задается формулой (15).

Отметим, что возможно рассмотреть ситуацию, когда отдельные серверы системы физически разделяют один и тот же ресурс. Пусть: $S = \{s_i\}$ – множество серверов в системе, разбитое на подмножества $S_j = \{s_i\}, \subseteq S$, такие, что $S_j \cap S_j = \emptyset, i \neq j$ и $\bigcup_j S_j = S$. В этом случае если ресурс задан своей кривой обслуживания, то, приняв модель разделения ресурса в виде планировщика задач, можно рассчитать ресурс, доступный для каждого из серверов подмножества S_j на общем ресурсе j [28]. Однако далее для упрощения предполагается, что серверы имеют фиксированный ресурс полностью задаваемой своей кривой обслуживания.

3.2. УРОВЕНЬ ПЛАТФОРМЫ

Рассмотрим уровень представления системы в виде платформы P, когда не выделяются сценарии ее использования, а в основном анализируется работа отдельных элементов. Примером может служить анализ производительности распределенной системы на уровне технических средств, отдельных компьютеров, коммутаторов и других устройств. Уровень платформы P по основным элементам аналогичен уровню представления для архитектуры системы, однако отличается используемой метрикой для оценки доступности. В отличие от уровня архитектуры, где анализируются временные характеристики потоков данных на пути обработки их в системе, для платформы оценка доступности связана с временными характеристиками отдельных узлов в системе.

Метрика задаётся отображением:

(19) $A_P: S \to D$

3.3. ФУНКЦИОНАЛЬНЫЙ УРОВЕНЬ

Функциональный уровень может быть описан в терминах уровней, приведенных выше. Его определение во многом зависит от задачи, поставленной при оценке доступности системы. Определим наиболее общие элементы функционального уровня и отношения между ними:

$\Phi = \{\phi_i\}$ – множество функций, выполняемых системой.

Для каждой функции определено множество компонентов уровней архитектуры (Σ) и платформы (P), участвующих в выполнении функции ϕ_i, через отображение:

(20) $Q: \Sigma \times P \to \Phi$

Декомпозиция системы на функциональном уровне связана с конкретными сценариями эксплуатации системы и может проводиться аналогично анализу дерева отказов в системе, например, по методике FMEA [6].

Метрика доступности A задаётся соответствующими метриками для компонентов более низкого уровня, в терминах которых описывается функция.

$A: \Phi \to D,$

Метрика сопоставляет для каждой из функций, выполняемых системой, некоторую величину, связанную с задержкой доступа к данной функции. В соответствии с уравнением (20), данная метрика может быть представлена как композиции для метрик более низкого уровня представления системы:

$$A = A_{\Sigma} \circ A_{P},$$

где A_{Σ}, A_{P} – метрики для архитектурного уровня и уровня платформы соответственно, через которые выражена заданная функция.

Оценка доступности задается уравнением (15).

Отметим, что подходы ТСИ можно двояко использовать в задаче оценки доступности. Первый сценарий применения ТСИ состоит в том, что метод используется для расчета максимальных значений временных параметров, которые потом будут использованы как ограничения в барьерной функции для расчета d_{max}.

Во втором сценарии по параметрам работающей системы рассчитываются максимально возможные задержки и делается вывод, удовлетворяет ли система ранее заданным ограничениям.

Оба сценария являются рабочими сценариями для практики, однако далее исследуется только второй сценарий; переход между сценариями не составляет трудностей.

Рассмотрим на макете программной системы пример оценки доступности для двух уровней референтной модели. Для оценки доступности выбрана относительно простая система, которую можно анализировать аналитически, что позволяет проследить подходы к реализации модели доступности с применением ТСИ.

4. Пример расчета доступности системы с последовательной обработкой данных

4.1. ОПИСАНИЕ СИСТЕМЫ

Рассмотрим систему без циклов в путях обработки данных (рис. 2).

Все серверы в системе реализуют одинаковую дисциплину обслуживания конкурентных потоков. Такое ограничение принято только для упрощения и наглядности приводимых выкладок.

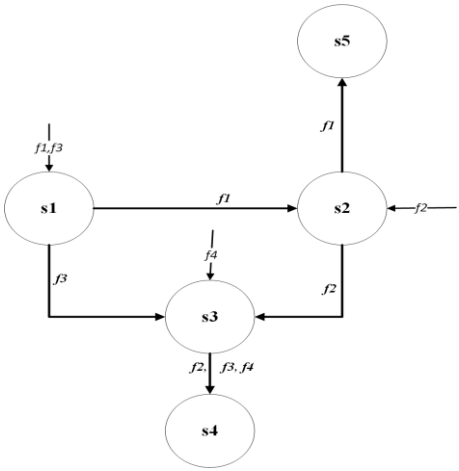


Рис. 2. Архитектура системы с путями потоков

Пусть в системе имеется пять серверов $S = \{s_1..s_5\}$, имеющих кривые обслуживания вида:

$$\beta_i^{R,T}(t) = \begin{cases} R(t-T), t > T, \\ 0. \end{cases}$$

В систему поступают четыре потока $F = f_1, \ldots, f_4$, огибающая потока i задана в виде кривой $\alpha_i(t) = rt + b$.

Так как некоторые режимы работы системы кроме расчетной модели проверялись на макете, то параметры модели выбраны с учетом ограничений используемых вычислительных средств для макетирования.

Так как адекватность макетирования во многом зависит от точности задания различных постоянных времени, которые для ОС с ядром Linux общего назначения составляет порядка 1 мс [14], то в соответствии с этим основные параметры производительности сервера и скорости потоков выбирались менее одного Кбайт. Для задержки T кривой обслуживания выбрана задержка порядка 1 с, что позволяло упростить отладку макета.

Параметры потоков и серверов обслуживания в примере заданы одинаковыми, так как пример служит для пояснения предлагаемой модели расчета доступности с помощью ТСИ, и одно-

образие параметров позволяет относительно просто продемонстрировать изменение доступности при варьировании параметров.

Зададим маршруты $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_4\}$, в системе для каждого из потоков для архитектурного уровня представления системы последовательностями ребер в графе G:

- $\mathcal{P}_1 = \{(s_1, s_2), (s_2, s_5)\}$,
- $\mathcal{P}_2 = \{(s_2, s_3), (s_3, s_4)\}$,
- $\mathcal{P}_3 = \{(s_1, s_3), (s_3, s_4)\}$,
- $\mathcal{P}_4 = \{(s_3, s_4)\}$.

Если необходимо указать огибающую потока на выходе определенного компонента, используется нотация $\alpha_i^j(t)$, где i – номер потока, к которому относится огибающая, j-номер компонента.

Для каждого уровня модели доступности приведен вид кривых обслуживания и огибающих потоков, необходимых для расчета задержки с применением формулы (6). Расчет максимальной задержки проводится в программе, написанной на языке Java на основе библиотеки для ТСИ [34]. Библиотека позволяет моделировать все три дисциплины обслуживания, для которых кривая обслуживания при совместной обработке потоков определялась теоремами 2–4 и выражениями (11)–(13) соответственно. Однако ниже приведены расчеты только дисциплины с произвольным выбором, так как результаты по другим дисциплинам не дают каких-либо принципиально новых данных о применимости модели оценки доступности на основе ТСИ.

В примерах моделировалась оценка доступности при изменении производительности компонентов системы и параметров потоков:

- Производительности всех серверов в системе. Такое поведение системы возможно, если, например, используются атаки на отказ в обслуживании, связанные с замедлением системы при обработке специально подобранных данных.
- Скорость входных потоков. Такое поведение системы возможно, если скорости потоков на входе меняются в результате DoS атаки.
- Неравномерность входных потоков.

Для апробации модели сделан макет системы в виде набора программ на языке С [12], реализующих сервер и задающих генераторы входных потоков, которые объединялись через сеть в моделируемую систему. Каждый сервер и каждый генератор потока оформлен в виде отдельного процесса ОС.

Для уменьшения влияния сетевых эффектов на измерения использовалась эмуляция сети в пределах одного компьютера и ограничивалась скорость и неравномерность потоков.

Программа генератора входных потоков работает в циклическом режиме, за фиксированный период в одну секунду посылается объем данных r, разбитый на посылки размера не более b.

Поток с необходимой огибающей генерировался по формуле $\alpha(t) = C\frac{t}{T} + B$, где T- период времени, в течение которого посылается объем данных размером C, упакованный в пакеты объемом не более чем B. Скорость потока $r = \frac{C}{T}$, неравномерность $b = B$. Чтобы учесть влияние работы системных процессов, для используемой ОС на основе ядра Linux проведено дополнительное моделирование, в котором оценивалась огибающая потока (рис. 3), поле чего опытном путем подбирались коэффициенты для скорости потока для соответствия теоретической огибающей.

Сервер реализован в виде программы с циклическим алгоритмом опроса каналов с вызовом системной функции *poll* [25].

Если ни на одном из каналов нет данных, то сервер останавливает опрос на время T. Для задания кривой обслуживания в сервере реализован алгоритм побайтной обработки поступающего пакета с задержкой между обработкой на время, обратно пропорциональное величине R. Порядок опроса каналов при совместной обработке нескольких потоков зависит от установленной дисциплины обслуживания. Для дисциплины с последовательной обработкой одновременно опрашиваются сразу все каналы, и если данные доступны по нескольким каналам, канал для обработки выбирается случайным образом один. Для дисциплины с произвольным выбором все каналы, обрабатываемые сервером, сканируются последовательно. Каждый новый цикл начинается со случайно выбранного канала, для которого выполняется вызов *poll* с ожиданием с временем T, и если данные присутствуют за это время, то они обрабатываются, а если нет, то

переходят к следующему каналу, который опрашивается без ожидания, после опроса всех каналов и обработки доступных данных цикл повторяется.

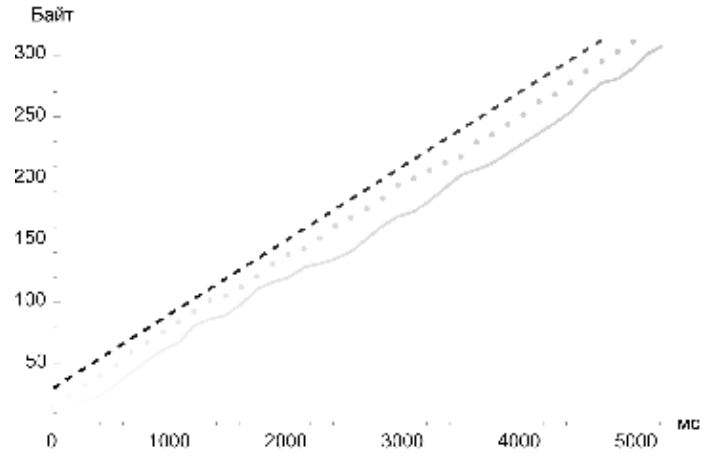


Рис. 3. Интегральный поток (сплошная линия), огибающая потока (пунктир), линейная аппроксимация (штрихованная линия) [1]*, для потока на выходе генератора сигнала. Теоретическая скорость потока r = 60 байт/с, неравномерность b = 30 байт*

Аналогично как для огибающей потока для калибровки программы оценивалась кривая обслуживания [13] и подбиралась скорость обработки данных на сервере, чтобы экспериментальная кривая обслуживания соответствовала теоретической минимальной с параметрами T, R. Минимальная теоретическая кривая обслуживания ограничивает снизу экспериментальную кривую обслуживания. Так как точная оценка минимальной кривой обслуживания на основе только экспериментальных данных является в общем случае нерешенной задачей, использовались приближенные методы [13] для оценки параметра R и T для конфигурации с единственным потоком на входе сервера. Далее полученные параметры R, T проверялись для системы с одни сервером

с несколькими потоками на входе, в результате экспериментальная кривая обслуживания получила резерв около 20%, который учитывает обработку данных ОС.

Более детально алгоритм генерации потока и его обработки описан в коде программ [12].

На макете измерялись те же параметры, что и рассчитывались в модели. В качестве значений выбиралось максимальное значение соответствующей задержки за время работы макета. В работе приведены данные эксперимента на макете за час работы. Выбор времени моделирования на макете обоснован практическими соображениями, «разумностью» затраченных ресурсов, а также тем, что большинство постоянных времени процессов в компьютерах ОС на базе ядра Linux не превышают десятков минут (а чаще значительно меньше).

4.2. АНАЛИЗ ДОСТУПНОСТИ НА УРОВНЕ ПЛАТФОРМЫ С ИСПОЛЬЗОВАНИЕМ ТСИ

Для оценки доступности используем интерпретацию теорем 2, 4, когда потоки, обрабатываемые на серверах, не разделяются, а рассматриваются как объединённый поток (TFA – Total Flow Analysis) [18], и ограничения по задержке вычисляются для объединённого потока на каждом из серверов по формуле:

$D = h(\alpha_{\sum}, \beta)$, где $\alpha_{\sum}$ – объединённая огибающая потоков на входе сервера.

Приведем формулы для расчета максимальной задержки только для серверов на пути одного потока. Задержки для серверов, которые входят в маршруты других потоков, но не входят в данный, могут быть рассчитаны аналогично.

Пример теоретических кривых обслуживания для каждого из компонентов на пути потока 2 (рис. 2) и объединённые огибающие потока на входах серверов приведены в таблице 1. Здесь $\beta_i^{lo,k}$ обозначает остаточную кривую обслуживания для компонента s_k, $\{s_k\} \subseteq S$. Для обозначения номера конкретного сервера для величин $\alpha_{\sum}, \beta$ добавляется надстрочный индекс с номером сервера, нижний индекс обозначает, для какого потока рассчитывается остаточная кривая обслуживания.

Таблица 1 Кривые обслуживания и огибающие потока относительно Потока 2 по методу TFA

Поток 2	
$\beta_1^{lo,1} = (\beta^1 \ominus (\theta_1^1, a_3^1))$	$a_\Sigma^1 = \alpha_3 + \alpha_1;$ $a_3^1 = \alpha_3$
$\beta_2^{lo,1} = \beta^2 \otimes (\ominus (\theta_1^2, a_2^2))$	$a_\Sigma^2 = \alpha_2 + a_1^2;$ $a_1^2 = \alpha_1 \oslash \beta_1^{lo,1}$ $a_2^2 = \alpha_2$
β^5	$a_\Sigma^1 = a_1^5$ $a_1^5 = a_1^2 \oslash \beta_2^{lo,1}$
β^3	$a_\Sigma^3 = \alpha_4 + a_2^2 + a_3^1;$ $a_3^1 = \alpha_3 \oslash \beta^1$ $a_2^2 = a_2^2 \oslash \beta^2$
β^5	$a_\Sigma^5 = (a_\Sigma^3 \oslash \beta^3)$

Рассмотрим оценку доступности на уровне платформы на основе измеренных и рассчитанных по ТСИ значений. Измерения и расчет проводились по всем четырём потокам с применением метода TFA для задержек по каждому из серверов; дополнительно по каждому из потоков указана полная задержка, рассчитанная по методу раздельного анализа потоков, описанному ниже. Примеры расчета задержки для дисциплины обслуживания с произвольным выбором на каждом из серверов приведены в таблице 2 вместе с измеренными на макете значениями.

Таблица 2. Сравнение задержки для ТСИ и макета системы. Параметры серверов системы: $R = 500$ *байт/с,* $T = 1$ *с, параметры входных потоков:* $r = 60$ *байт/с,* $b = 30$ *байт*

Задержка по серверу	Поток 1 (d_1 с)		Поток 2 (d_2 с)		Поток 3 (d_3 с)		Поток 4 (d_4 с)	
	ТСИ	Экс.	ТСИ	Экс.	ТСИ	Экс.	ТСИ	Экс.
Сервер 5	1,3	**1,0**						
Сервер 4			2,9	**2,1**	2,9	1,9	2,9	1,9
Сервер 3			2,3	**1,8**	2,3	1,8	2,3	1,7
Сервер 2	1,7	1,3	1,7	1,4				
Сервер 1	1,5	1,2			1,5	**1,3**		

Для оценки стабильности работы системных процессов эксперименты с измерением задержки на макете были повторены 10 раз, и в таблице 2 приведены средние значения, максимальное значение доверительного интервала с доверительной вероятностью 0,95 для измеренных на серверах задержек составляет 86 мс.

Как видно из таблицы, рассчитанные задержки больше измеренных экспериментальных задержек для отдельных потоков, что является следствием того, что для расчета использовалась минимальная кривая обслуживания (6). Для того чтобы рассматривать результаты измерений в рамках метода TFA, необходимо использовать максимальное значение из задержек каждого из потоков, обрабатываемых на сервере. Максимальные из измеренных на серверах задержки выделены в таблице жирным шрифтом, различие в расчетном значении по ТСИ для минимальной кривой обслуживания и измеренном на макете не превышает 30%. Все измеренные задержки находятся в рамках результатов ТСИ, которая ограничивает значение задержки сверху. Относительное расхождение объясняется особенностями передачи данных в ядре ОС, которые сложно учесть в макете, и наличием редких событий, увеличивающих задержку, например одновременного прихода больших пакетов данных для каждого из потоков, обрабатываемых сервером, регистрация которых требует значительного увеличения времени эксперимента.

4.3. АНАЛИЗ ДОСТУПНОСТИ НА АРХИТЕКТУРНОМ УРОВНЕ С ИСПОЛЬЗОВАНИЕМ ТСИ

Оценка доступности на архитектурном уровне требует более высокой степени детализации системы. Например, это может быть выражено в том, что для каждого из потоков $\{f_1, \ldots, f_4\}$ по всем серверам, входящим в его маршрут, оценивается отдельно кривая обслуживания (теоремы 2 и 3 в зависимости от реализуемой дисциплины обработки) и для потока оценивается общая кривая обслуживания по всей цепочке серверов (правило 1).

В ТСИ такой подход известен как раздельный анализ SFA (separate flow analysis), когда для фоновых потоков вычисляется общая доля потребляемых ресурсов, которая потом позволяет вычислить по теореме 3 общий остаточный ресурс (общую кривую

обслуживания) β^{lo} для анализируемого потока и на основе β^{lo} далее вычислить ограничения на максимальную задержку. Наличие общей кривой обслуживания по потоку позволяет использовать принцип «платить за выбросы один раз» (pay burst only once) [32] и получить лучшую оценку для максимальной задержки, Заметим, что рассчитанные значения максимальной задержки с применением принципа «платить за выбросы однажды» могут отличаться от интегральной задержки, полученной суммированием отдельных задержек по пути потока.

При использовании ТСИ существует возможность различной последовательности применения теорем 2, 4 и правила 1, заключающаяся в том, в какой последовательности выполнять операции $\ominus$, $\otimes$ при расчете кривой обслуживания для потока. Известно, что когда приоритет отдается операции объединения кривых обслуживания перед применением оператора $\ominus$, принцип «платить за мультиплексирования только раз» (PMO – Play Multiplexing Only Once) позволяет получить лучшую оценку для максимальной задержки в системе [18]. Однако последний принцип возможен только для систем, имеющих вложенную топологию (nested networks) [38], т.е. когда отдельные потоки в пределах пути не разъединяются, чтобы потом соединиться. Поэтому результаты приведем без учета данного принципа.

Рассмотрим применение методов ТСИ для расчета доступности в исследуемом макете. Анализ ограничений по кривым обслуживания и огибающие потоков по пути для каждого из потоков приведены в таблице 3. Здесь для $\beta_k^{lo}(t)$ нижний индекс обозначает сервер s_k.

Для сравнения расчетных и измеренных значений воспользуемся измерениями, сделанными для уровня платформы (таблица 2), и проведём суммирования по задержкам для серверов на пути каждого из потоков и сравним с результатами по SFA (таблица 4), Экспериментальные значения отличаются менее чем на 20% от расчетных для минимальной теоретической кривой обслуживания.

Таблица 3. Кривые обслуживания и огибающие потоков

Остаточная кривая обслуживания	Огибающая потока
Поток 1	
$\beta_1^{lo} = \beta^1 \otimes (\beta^2 \ominus (\theta_2^2, a_2^2)) \otimes \beta^4$	$a_2^2 = \alpha_2;$
Поток 2	
$\beta_2^{lo} = \beta^2 \otimes ((\beta^3 \ominus (\theta_4^3, a_4^3)) \ominus (\theta_3^3, a_3^3)) \otimes ((\beta^4 \ominus (\theta_4^4, a_4^4)) \ominus (\theta_3^4, a_3^4))$	$a_4^3 = \alpha_4$ $a_3^3 = a_3^1 \oslash (\beta^1 \ominus (\theta_1^1, a_1^1))$ $a_3^4 = = a_3^1 \oslash (\beta^1 \ominus (\theta_1^1, a_1^1)) \otimes ((\beta^3 \ominus (\theta_2^3, a_2^3)) \ominus (\theta_4^3, a_4^3))$ $a_4^4 = \beta^3 \otimes \alpha_4$
Поток 3	
$\beta_3^{lo} = (\beta^1 \ominus (\theta_1^1, a_1^1)) \otimes ((\beta^3 \ominus (\theta_2^3, a_2^3)) \ominus (\theta_4^3, a_4^3)) \otimes ((\beta^4 \ominus (\theta_2^4, a_2^4)) \ominus (\theta_4^4, a_4^4))$	$a_1^1 = \alpha_1$ $a_4^3 = \alpha_4$ $a_2^3 = a_2^2 \oslash (\beta^2 \otimes ((\beta^3 \ominus (\theta_4^3, a_4^3)) \ominus (\theta_3^3, a_3^3))$ $a_4^4 = a_4^3 \oslash \beta^3$
Поток 4	
$\beta_4^{lo} = \beta^3 \otimes \beta^4$	α_4

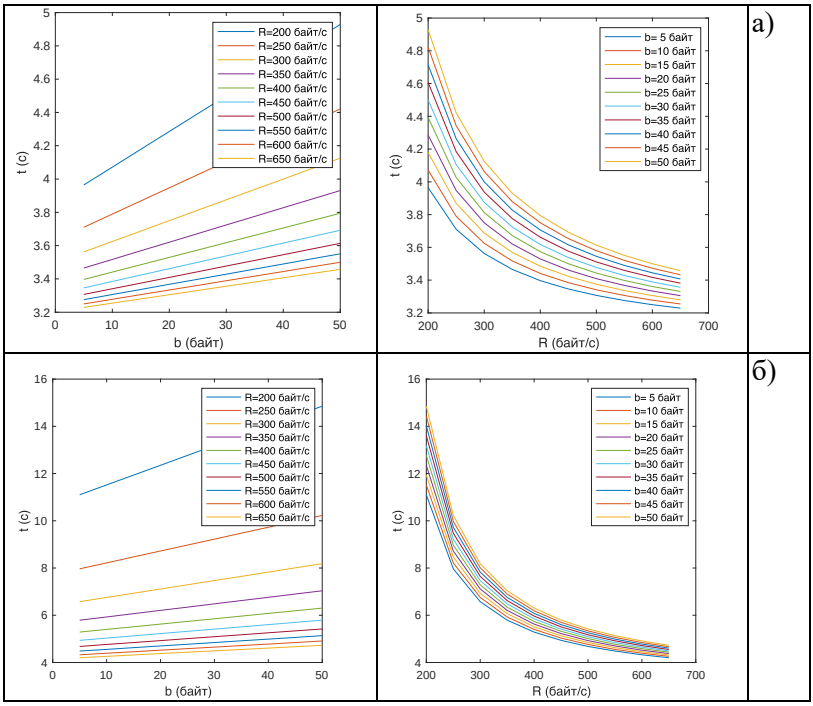
Таблица 4. Сравнение задержки по потоку для ТСИ и макета системы, $R = 500$ байт/с, $T = 1$ с, параметры входных потоков: $r = 60$ байт/с, $b = 30$ байт. На серверах реализована дисциплина обработки с произвольным выбором

Полная задержка по потоку (SFA)	Поток 1(d_1 с)		Поток 2(d_2 с)		Поток 3(d_3 с)		Поток 4(d_4 с)	
	ТСИ	Экс,	ТСИ	Экс,	ТСИ	Экс,	ТСИ	Экс,
	3,5	3,5	5,1	5,3	5,0	5,0	4,2	3,6

Экспериментальная задержка по второму потоку оказалась несколько выше, чем значение ТСИ, что отражает большое влияние ядра ОС и ошибку измерений на каждом из серверов по пути потока с учетом, что пути для потоков f_2, f_3 самые «длинные» в системе и начальный сервер s_3 для потока f_2 , в отличие от потока f_3, обрабатывается совместно с уже возмущенным после обработки потоком f_1 с увеличенной неравномерностью.

Для архитектурного уровня рассмотрим также оценку доступности при изменении параметров системы или входных потоков. На рис, 4, 5 показаны изменения максимальной задержки для потоков 1-4 для произвольной дисциплины обслуживания на серверах.

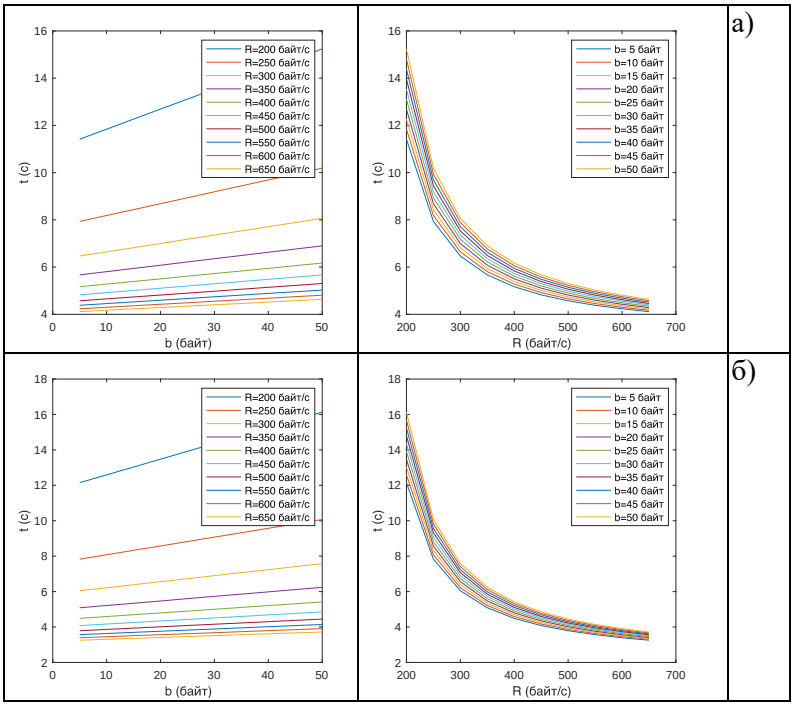
Для графика варьировалась производительность серверов и неравномерности потоков в диапазоне $R = 200 - 650$ байт/с, $T = 1$ с, параметры входных потоков: $r = 60$ байт/с, $b = 5 - 50$ байт.



Рис, 4, Изменение максимальной задержки для потоков 1–2 – а), б) соответственно – от b (слева) и R (справа) при фиксированных R и b

Как видно из графиков, общий характер зависимости максимальной задержки от неравномерности потоков линейный и обратно пропорциональный производительности сервера. Первый

поток наименее чувствителен (рис. 4) к моделируемым изменениям системы с точки зрения задержки прохождения данных, так как он имеет относительно короткий маршрут обработки в системе и мало взаимодействует с другими потоками. Остальные потоки в исследуемом диапазоне изменения параметров имеют пороговое R значение, при котором начинается резкий рост задержки прохождения данных, что связано с тем, что у части серверов загрузка приближается к критической. Такое поведение для реальной системы необходимо учитывать при оценке доступности, например установления пороговых значений для барьерной функции.



Рис, 1, Изменение максимальной задержки для потоков 3–4 – а), б) соответственно – от b (слева) и R (справа) при фиксированных R и b

4.4. АНАЛИЗ ДОСТУПНОСТИ НА ФУНКЦИОНАЛЬНОМ УРОВНЕ

Как указано выше, анализ на функциональном уровне зависит от сценариев применения системы и целей, поставленных при анализе доступности. В качестве входных данных могут быть использованы результаты для оценки максимальной задержки, полученные как на архитектурном уровне, так и на уровне платформы. Анализ доступности может рассматриваться как набор ограничений на время, которое требуется на обработку информации по нескольким маршрутам потоков и/или для набора серверов, задействованных в реализации конкретной функции.

Например, в случае анализа на уровне архитектуры интегральная задержка для всего может быть равна сумме задержек на компонентах, принимающих участие в выполнении функции:

$D_i = \sum_{j \in S} d^j$.

Если функциональный анализ проводится на архитектурном уровне, то возможный сценарий на установку ограничений для максимальной задержки как суммы максимальных задержек потоков данных, используемых для выполнения функции:

$D_i = \sum_{j \in \mathcal{P}_i} d^j$.

5. Обсуждение и заключение

Проблеме проектирования и анализа цифровых систем с заданными параметрами доступности с технической точки зрения не всегда уделялось достаточное внимание, несмотря на то, что доступность является приоритетной целью для многих промышленных систем управления [9], например, при реализации принципа безопасного проектирования и обеспечения информационной безопасности. Одной из причин недостаточного внимания являлось отсутствие практических методов ее оценки для компьютерных систем. В работе предложена референтная модель доступности для анализа систем, призванная решить данную проблему. Модель задает три уровня, на которых анализируется доступность в системе, а также определяет набор вспомогательных методов, необходимых для применения модели.

В работе в качестве основы для реализации вспомогательных методов предложено использовать теорию сетевых исчислений (ТСИ). Ее применение логично с точки зрения получаемых на выходе ТСИ значений для максимальной задержки обработки данных в системе и применения их в референтной модели доступности. Применение ТСИ для оценки доступности не является единственно возможным методом реализации предложенной референтной модели. Например, альтернативой могут служить и методы теории массового обслуживания (ТМО), и просто прямые измерения, если система доступна для анализа. Преимуществом ТСИ является уход от необходимости статистического описания характеристик системы, которое необходимо, например, в случае ТМО и его замены более простыми в инженерном плане детерминированными ограничениями на потоки и компоненты системы, ТСИ и ТМО во многом дополняют друг друга. Кроме детерминированного подхода ТСИ, существует ее стохастическая интерпретация [20, 27]. Однако наш опыт показывает, что в практических приложениях использование стохастической ТСИ не всегда оправдано, так как она лишает ТСИ прозрачности вычислений и независимости от знания статистических характеристик системы и в инженерном плане имеет схожие ограничения, что и ТМО.

Для более точного описания различных дисциплин с совместной обработкой потоков в ТСИ разработаны соответствующие кривые обслуживания. В работе предложено расширение кривой обслуживания при последовательной обработке сигналов на дисциплину с последовательной обработкой сигнала с ожиданием и блокированием работы при отсутствии данных на входе.

Недостатками ТСИ, уже указанными ранее, является условие отсутствия потерь данных или их генерации внутри системы, что на практике сужает класс компьютерных систем, где может применяться ТСИ. Частично данное ограничение обходится в развитии ТСИ для потоков с переменным масштабом [24], однако их применение должно быть основано на детальном знании внутренних алгоритмов работы системы.

Серьёзным ограничением для ТСИ является ее ориентированность на системы, где отсутствуют циклические зависимости

как внутри потока, так и между потоками. Необходимость выделения систем с циклической зависимостью определяется трудностями обеспечения сходимости алгоритмов при расчете огибающих потоков и остаточной кривой обслуживания. Данная проблема еще не нашла общего решения и требует дальнейшего внимания.

Интерпретация референтной модели доступности на основании положений ТСИ позволило решить задачу оценки доступности для распределенных цифровых систем с конкурентными потоками данных и предложить реализацию, адаптированную для инженерных расчетов.

Предложенная в работе трехуровневая референтная модель доступности акцентирует внимание на том, что доступность должна анализироваться, по возможности, на всех уровнях референтной модели, так как оценка доступности на функциональном уровне является композицией результатов, полученных для архитектурного уровня и/или уровня платформы референтной модели.

В примере оценки доступности, приведенном в работе, рассматривалось влияние изменения производительности компонентов на задержку в обработке потока данных на компонентах системы. Понижение загрузки отдельных компонент из-за изменения производительности или уменьшения неравномерности потоков ожидаемо приводит к улучшению максимальной задержки как по потокам на системном уровне, так и для локальных характеристик по задержке на компонентах. Однако изменение задержки при малой производительности системы быстрее проявляется на потоках, чей путь наиболее пересекается с другими потоками в системе.

Апробация применения референтной модели доступности на основе ТСИ с использованием макета показало, что эксперимент не противоречит теории и измеренная задержка не превышает предельных характеристик, рассчитанных с помощью ТСИ. Как и ожидалось, ТСИ дает максимальную задержку большую, чем удалось измерить на макете, так как при расчете в качестве кривой обслуживания принята минимальная кривая обслуживания. В среднем различие измеренной и рассчитанной задержки не пре-

вышает 30%. Однако следует отметить, что соответствие рассчитанной и измеренных максимальных задержек могло варьироваться в зависимости от дисциплины обслуживания и от эксперимента к эксперименту в пределах сотни мс. Одним из источников вариативности при измерениях на макете является то, что в отличие от модели, рассчитываемой аналитически, в макете поток между серверами претерпевает дополнительную обработку в сетевом ядре операционной системы, учет влияния которого является трудной задачей. Создание программной реализации сервера ТСИ с соответствующей дисциплиной наталкивается на трудности неоднозначности интерпретации минимальной кривой обслуживания, когда фактически отпрядены только нижние границы производительности сервера, между тем обратная задача описания реальной системы решается удовлетворительно [29, 33]. Рассмотренный в работе пример системы для наглядности имеет небольшой размер, однако иерархическая структура модели с одной стороны и масштабируемость ТСИ [26, 36] – с другой дает возможность применить предложенную модель для систем с большим количеством компонентов.

Референтная модель доступности и ее интерпретация в ТСИ имеет более широкое применение чем только в задачах информационной безопасности. Предложенную модель можно применять для синтеза и обоснования архитектуры распределенных модульных систем совместно с другими моделями и методами синтеза систем [11].

Литература

1. БАЙБУЛАТОВ А.А., ПРОМЫСЛОВ В.Г. *Аппроксимация огибающей в приложениях «Network calculus»* // Проблемы управления. – 2016. – №6. – С. 59–64.
2. БАЙБУЛАТОВ А.А., ПРОМЫСЛОВ В.Г. *О свойстве доступности и его метрике для АСУ ТП АЭС* // Труды 15-й Международной конференции «Управление развитием крупномасштабных систем» (MLSD’2022). – М.: ИПУ РАН, 2022. – Т. 3. – С. 1020–1024.
3. БАХАРЕВА Н.Ф., ТАРАСОВ В.Н. *Аппроксимативные методы и модели массового обслуживания, Исследование компьютерных сетей*. – Самара: Изд-во СНЦ РАН, 2017. – 328 с.

4. ВИШНЕВСКИЙ В.М. *Теоретические основы проектирования компьютерных сетей*. – М.: Техносфера, 2003. – 512с.
5. ВИШНЕВСКИЙ В.М., ГОРБУНОВА А.В. *Применение методов машинного обучения к решению задач теории массового обслуживания* // ИТиВС. – 2021. – №4. – С.70–82.
6. *ГОСТ Р 51901,12-2007 (МЭК 60812:2006) Метод анализа видов и последствий отказов.*
7. *ГОСТ Р ИСО /МЭК 15408-1-2008 Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий*. Часть 1. Введение и общая модель.
8. *ГОСТ Р 27,002-2009 Национальный стандарт Российской Федерации. Надежность в технике. Термины и определения*.
9. *ГОСТ Р 56205-2014 IEC/TS 62443-1-1:2009 Защищенность (кибербезопасность) сети и системы*. Часть 1-1. Терминология, концептуальные положения и модели.
10. КАЛАШНИКОВ А.О., АНИКИНА Е.В. *Управление информационными рисками сложной сети на основе метода стохастического имитационного моделирования (часть 1)* // Информация и безопасность. – 2019. – Т. 22, Вып. 1. – С. 6–13.
11. КУЗНЕЦОВ Н.А., КУЛЬБА В.В., КОВАЛЕВСКИЙ С.С. и др. *Методы анализа и синтеза модульных информационно-управляющих систем*. – М.: Физматлит, 2002. – 800 с.
12. *Программы для эмуляции сервера и задающего генератора потока на языке С.* – URL: https://www.dropbox.com/scl/fo/txbbdux1nr89c6unr9yi7/h?rlkey=h6hc6y36a1h6r9ex0kwf8iejn&dl=0 (дата обращения: 12.01.2024).
13. ПРОМЫСЛОВ В.Г., СЕМЕНКОВ К.В., ЖАРКО Е.Ф. *Оценка собственной характеристики киберфизической системы методом сетевых исчислений* // Управление большими системами. – 2023. – Вып. 105. – С. 6–29.
14. ADAM G. *Real-Time Performance and Response Latency Measurements of Linux Kernels on Single-Board Computers* // *Computers*. – 2021. – Vol. 10(64). – DOI: 10.3390/computers10050064.
15. BAYBULATOV A.A., PROMYSLOV V.G. *Industrial Control System Availability Assessment with a Metric Based on Delay and Dependency* // IFAC-PapersOnLine. – Elsevier, Amsterdam, 2021. – Vol. 54, Iss. 13. – P. 472–476.
16. BAYBULATOV A.A., PROMYSLOV V.G. *A Metric for the IACS Availability Risk Assessment* // Proc. of the Int. Russian Automation Conf.

(RusAutoCon), Sochi: IEEE, 2022. – P. 750–754. – URL: https://ieeexplore,ieee,org/document/9896250.
17. BECK M., SCHMITT J. *The disco stochastic network calculator version 1.0: When waiting comes to an end* // Proc. Valuetools. – 2013. – P. 282–285.
18. BONDORF S., SCHMITT J. *The DiscoDNC v2 – A Comprehensive Tool for Deterministic Network Calculus* // EAI Endorsed Trans. on Internet of Things. – 2014. – Vol. 1. – DOI: 10.4108/icst.valuetools.2014.258167.
19. CHAKRABORTY S., SIMON K., THIELE L. *A general framework for analysing system properties in platform-based embedded system designs* // Proc. of the Conf. on Design, Automation and Test in Europe – Vol. 1, DATE'03. – Washington, DC, USA, IEEE Computer Society, 2003. – P. 10190–10195
20. CHANG C.-S. *Stability, queue length and delay, ii, stochastic queueing networks* // Proc. IEEE CDC. – 1992. – Vol. 1. – P. 1005–1010.
21. CHEN Q., ABERCROMBIE R. SHELDON F. *Risk Assessment For Industrial Control Systems Quantifying Availability Using Mean Failure Cost (MFC*) // Journal of Artificial Intelligence and Soft Computing Research. – 2015. – Vol. 5. – P. 205-220. – DOI: 10.1515/jaiscr-2015-0029.
22. CRUZ R.L. *A Calculus for Network Delay, Part I: Network Elements in Isolation* // IEEE Trans. on Information Theory. – Jan. 1991. –Vol. 37. – P. 114–131.
23. CRUZ R.L. *A Calculus for Network Delay, Part II: Network Analysis Information Theory* // IEEE Trans. on Information Theory. – Jan. 1991. – Vol. 37. – P. 132–141.
24. FIDLER M., SCHMITT J. *On the way to a distributed systems calculus: An end-to-end network calculus with data scaling* // Proc. of the Joint Int. Conf. on Measurement and Modeling of Computer Systems. – P. 287–298. – DOI: https://doi.org/10.1145/1140277.
25. *ISO/IEC/IEEE 9945:2009 Information technology Portable Operating System Interface (POSIX®) Base Specifications*. – Iss. 7.
26. JACOBS N., HOSSAIN-MCKENZIE S., SUMMERS A. *Modeling Data Flows with Network Calculus in Cyber-Physical Systems: Enabling Feature Analysis for Anomaly Detection Applications* // Information. – 2021. – Vol. 12. – P. 255. – DOI: https://doi.org/10.3390/info12060255.
27. JIANG Y., LIU Y. *Stochastic Network Calculus*. – Springer, 2008.

28. JONSSON B., PERATHONER S., THIELE L. et al. *Cyclic dependencies in modular performance analysis* // Proc. of the 8th ACM Int. Conf. on Embedded software, EMSOFT'08, New York, NY, USA, 2008, ACM. – P. 179–188.
29. KUROSE J. *On computing per-session performance bounds in high-speed multi-hop computer networks* // Proc. of ACM SIGMETRICS. – 1992. – P. 128–139.
30. FLUCHS S., TAŞTAN E., TRUMPF T. et al. *Traceable Security-by-Design Decisions for Cyber-Physical Systems (CPSs) by Means of Function-Based Diagrams and Security Libraries* // Sensors. – 2023. – Vol. 23. – P. 5547. – DOI: https://doi.org/10.3390/s2312554.
31. GEISMANN J., BODDEN E. *A systematic literature review of model-driven security engineering for cyber–physical systems* // Journal of Systems and Software. – 2020. – Vol. 169. – P. 110697. – DOI: https://doi.org/10.1016/j.jss.2020.110697.
32. LE BOUDEC J.-Y., THIRAN P. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Online Version of the Book Springer Verlag, – LNCS 2050. Version April 26, 2012. – 263 p.
33. LUIGI A., GIUSEPPE B., D'ACQUISTO G. *Service curve estimation by measurement: an input output analysis of a softswitch model* // Proc. of the Third Int. Conf. on Quality of Service in Multiservice IP Networks (QoS-IP'05), 2005. – Springer-Verlag, Berlin, Heidelberg. – P. 49–60. – DOI: https://doi.org/10.1007/978-3-540-30573-6_4.
34. *NetworkCalculus,org DNC (NCorg DNC).* – URL: https://github.com/NetCal/DNC (дата обращения: 12.12.2023).
35. https://www.cl.cam.ac.uk/teaching/1011/R01/75-protection.pdf (дата обращения: 30.03.2024).
36. SCHMITT J.B., ZDARSKY F.A., MARTINOVIC I. *Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once* // GI/ITG MMB, 2008.
37. SIMON I. *Recognizable sets with multiplicities in the tropical semiring* // Mathematical Foundations of Computer Science. Lecture Notes in Computer Science. – 1988. –Vol. 324. – P. 107–120.
38. SCHEFFLER A., BONDORF S., SCHMITT J. *Analyzing FIFO-Multiplexing Tandems with Network Calculus and a Tailored Grid Search* // The 34th Int. Teletraffic Congress (ITC-2022), 2022.

AVAILABILITY MODEL BASED ON NETWORK CALCULUS FOR DATA FLOW PROCESSING SYSTEM

Vitaly Promyslov, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Cand,Sc, (vp@ipu,ru).

Abstract: The work examines the problem of assessing accessibility in digital computing systems focused on flow data processing, Availability is considered in the context of the confidentiality, integrity, availability (CIA) model of information security, Availability is characterized by the fact that it is a "point" assessment of the time characteristics of a system, its function or element, To assess availability, a three-level reference model is proposed, associated with different representations of the system at the architectural and functional levels, The formulation of the model is considered within the framework of the network calculus theory (NC), which makes it possible to apply this model to calculate the accessibility of digital computing systems in practice, To test the model, the work analyzes the availability of the system on a mock-up of a digital computing system with competitive processing of streaming data on the server, The possibility of calculating system parameters for various disciplines of processing competing threads on the server is shown.

Keywords: availability, model, design, analysis, network calculus, information security.

УДК 021.8 + 025.1
ББК 78.34
DOI: 10.25728/ubs.2024.110.5

Статья представлена к публикации членом редакционной коллегии В.Г. Лебедевым.

Поступила в редакцию 26.02.2024.
Опубликована 31.07.2024.