УДК 004.94(07)

Графо-аналитический метод идентификации аномального состояния распределенных информационных систем

Буравлев А. С., Васильев Н. В., Демидова Д. Е., Будко Н. П.

Постановка задачи: *Современные распределённые информационные системы отличаются высокой сложностью, масштабируемостью и требованиями к отказоустойчивости, что затрудняет их диагностику. Существующие методы идентификации аномального состояния таких систем используют для установления факта возникновения инцидента журналы функционирования отдельных узлов. Однако лежащий в их основе методологический базис не обеспечивает в полной мере локализацию сбоя в структуре распределенной информационной системы и не позволяет осуществлять анализ динамики его развития. В настоящей работе предложен метод идентификации аномальных состояний с применением графо-аналитического подхода для моделирования и анализа их динамики развития в распределенных информационных систем. Для построения моделей состояния распределенной информационной системы во времени («снимков») используется объединенный журнал, собираемый с её компонентов (узлов) и описывающий последовательность совмещенных программных вызовов в процессе вычисления и коммуникации. Метод включает в себя этап очистки и унификации журналов компонент распределенной информационной системы, этап извлечения из журнальных записей метрик использования вычислительных ресурсов, этап моделирования поведения распределенной информационной системы в виде последовательностей взвешенных графов и собственно этап идентификации аномальных состояний на основе кластеризации построенных на предыдущем этапе графов методом k-средних по мере расстояния редактирования. Каждый из кластеров соотносится с одним из состояний распределенной информационной системы. Как результат кластеризации также определяются усреднённые модели «типичного» поведения системы в каждом из состояний. Соотнесение каждого из графов поведения с одним из кластеров позволяет выделить интервалы аномальной активности и моменты переходов между нормальными и аварийными состояниями. Дальнейший ретроспективный анализ взвешенных графов на выделенных участках позволяет углубить понимание происходящих процессов.* ***Используемые методы:*** *методы теории графов; методы анализа данных; технологии извлечения процессов; методы системного анализа.* ***Новизна работы:*** *новизна исследования определяется применением для идентификации аномального состояния графо-аналитических методов, их уточнением для применения в анализе журналов распределенных информационных систем.* ***Результат:*** *описанный в работе метод реализован в виде прототипа программного комплекса, обеспечивающего визуализацию динамики поведения распределенной информационной системы в виде графов, анализ изменения их структуры во времени (расстояние редактирования), а также идентификацию во времени моментов аномального поведения.* ***Практическая значимость:*** *результаты показали, что метод позволяет выявлять трудновоспроизводимые ошибки и отслеживать развитие перегрузки узлов, сбои и нарушения синхронизации. В отличие от существующих решений, предложенный подход обеспечивает комплексный анализ объединённых журналов с нескольких узлов без модификации кода приложения, что подтверждает его применимость для промышленных распределённых информационных систем.*

Ключевые слова: *журнал событий, кластеризация графа, мониторинг, распределенная информационная система, расстояние редактирования графа, средний граф, трассировка.*

Библиографическая ссылка на статью:
Буравлев А. С., Васильев Н. В., Демидова Д. Е., Будко Н. П. Графо-аналитический метод идентификации аномального состояния распределенных информационных систем // Системы управления, связи и безопасности. 2025. № 4. С. 179-199. DOI: 10.24412/2410-9916-2025-4-179-199

Reference for citation:
Buravlev A. S., Vasiliev N. V., Demidova D. E., Budko N. P. Graphical-analytical method for identifying abnormal conditions in distributed information systems. *Systems of Control, Communication and Security*, 2025, no. 4, pp. 179-199 (in Russian). DOI: 10.24412/2410-9916-2025-4-179-199

Введение

Согласно ГОСТ 34.321-96 [1] распределенная информационная система (РИС) – информационная система, объекты данных и/или процессы которой физически распределяются на две или более компьютерные системы. Традиционно, процесс отладки таких систем основан на анализе журналов работы ее отдельных компонентов, расположенных на отдельных серверах. И если при небольшом количестве узлов такой подход позволяет выявлять ошибки, то в условиях десятков и сотен узлов он становится малоприменимым вследствие высокой вероятности возникновения трудно диагностируемых аномальных состояний связанных с ошибками параллельной обработки, синхронизации и ненадежностью каналов связи. Имеющиеся в настоящее время исследования в области идентификации аномалий поведения в телекоммуникационных сетях связи [2, 3] можно разделить на три основные группы:

- *сигнатурные методы*. Основаны на сопоставлении наблюдаемой активности с заранее известными шаблонами аномального поведения. Методы данной группы обеспечивают высокую точность для известных угроз, однако не позволяют обнаруживать новые, ранее не встречавшиеся типы аномального поведения;
- *статистические методы*. Основаны на статистическом анализе агрегированных метрик трафика и состояния узлов (энтропия, интенсивность потоков, отклонения параметров трафика). Данные методы главным образом фиксируют факт отклонения, не обеспечивая детализации причин возникновения аномалии и не локализуя компонент, инициировавший сбой;
- *методы на основе машинного обучения.* В настоящее время данная группа представлена наиболее широко, – от логико-вероятностных подходов, использующих байесовские сети и алгоритмы ассоциативных правил до методов на основе нейронных сетей и обучения с подкреплением.

Перечисленные методы осуществляют анализ трафика или отдельных узлов системы изолированно без учета структурных особенностей распределённой информационной системы в динамике ее функционирования. Как правило, за кадром остается программная логика РИС. А на стадии отладки и ввода системы эксплуатации возможность «заглянуть под капот» работающей распределенной информационной системы в момент отказа бывает критически важной. Существующие же методы способны обнаружить факт отклонения от нормы, выявить первопричину в потоке отказов, но не позволяют реконструировать последовательность произошедших в РИС событий.

С технологической точки зрения, существующие программные комплексы анализа состояния РИС обычно рассматривают отдельные узлы как «чёрные ящики», ограничиваясь аппаратными метриками (нагрузка процессора, использование памяти, диска). При таком подходе выявление реальных причин отказов, особенно если они вызваны проблемами в синхронизации параллельных

программных компонентов, является трудной задачей. Дальнейшее развитие технологий диагностики и анализа состояний шло в сторону разработки детальных механизмов фиксации поведения отдельных программных компонентов РИС в журналах. В частности, действующая рекомендация OpenTelemetry предполагает, что на каждом узле формируется журнал событий, включающий структурированные записи, автоматически генерируемые программными компонентами в процессе их функционирования. Такие журналы содержат сведения о потоке выполнения, типе операции или вызова, времени возникновения события и т. д. Объединенный журнал может быть основой для реконструкции модели поведения РИС, включая межузловые взаимодействия. Такая комплексная модель позволяет идентифицировать аномальные или переходные состояния. Одним из способов получения комплексной модели поведения РИС является использование методов глубинного анализа процессов (process mining), позволяющих восстанавливать модель состояния системы по журналам событий. Поскольку получаемые результирующие модели имеют графовую структуру, их дальнейший анализ естественным образом переходит в область графо-аналитических методов.

В данной работе предпринята попытка разработки и натурного моделирования метода идентификации аномального состояния РИС, использующего графо-аналитическое моделирование происходящих в системе информационно-телекоммуникационных процессов с целью определения временных интервалов возникновения и завершения аномального состояния. В качестве основного источника данных метод использует журнал событий, получаемый в результате объединения журналов функционирования компонентов РИС. Метод предполагает реконструкцию и использование трех моделей:

– графа совместного усредненного поведения взаимодействующих компонентов РИС на выбранном временном интервале;
– временной диаграммы изменения метрики расстояния редактирования графов совместного усредненного поведения взаимодействующих компонентов на последовательных временных интервалах;
– графа типичного поведения взаимодействующих компонентов РИС из выбранного кластера, соответствующего определенному состоянию системы.

Использование указанных моделей позволяет идентифицировать и локализовать периоды аномального поведения в объединенных журналах, что упрощает дальнейший анализ вызвавших это поведение причин.

В последующих частях работы приведены основные обозначения и терминологический аппарат; рассмотрен анализ методов и инструментов отладки распределённых информационных систем, а также их преимущества и недостатки; представлено описание метода, включая его основные этапы, а также выявление интервалов переходных процессов; доведены результаты экспериментального исследования разработанного метода на развернутом фрагменте корпоративной распределенной информационной системы в режиме пиковой нагрузки и при штатном её состоянии.

Основные обозначения и терминологический аппарат

При изложении граф-аналитического метода идентификации аномального состояния распределенных информационных систем и анализа их поведения вводятся следующие условных обозначения, показанные в таблице 1.

Таблица 1 – Основные обозначения

Обозначение	Физический смысл обозначения
$G = (V, E, \alpha, \beta)$	– Граф, описывающий набор связанных трасс узлов распределенной системы и представляемый: множеством вершин V, множеством дуг E, векторами параметров узлов α и ребер β
$\hat{G} = \left(\hat{V}, \hat{E}, \hat{\beta}, \hat{\delta}\right)$	– Средний граф представляемый: усредненным по частоте множеством вершин $\hat{V}$, множеством соединяющих усредненные вершины дуг $\hat{E}$, вектором $\hat{\beta}$ – средний вес ребер (загрузка CPU – Central Processing Unit) вектором $\hat{\delta}$ – средний вес вершин (загрузка оперативной памяти MEM – Memory)
$d(G, G')$	– Расстояние редактирования между графами
$\mu=\{G_1, G_2, \ldots, G_n\}$	– Средний граф последовательности графов с минимальным суммарным расстоянием редактирования до других графов
$\{C_1, C_2, \ldots, C_k\}$	– Набор кластеров графов, характеризующих k состояний исследуемой распределенной системы (по одному кластеру на каждое состояние)
$[t, t+1]$	– Временной интервал наблюдения для формирования графа состояния
$G_1^{\max} = \{\alpha, \beta\}$	– Максимально общий подграф графов g и g_2 (*maximal common subgraph – MCS*)
$\lvert G \rvert$	– Суммарное число вершин и ребер в графе G
S_i	– i-е состояние распределенной системы
GED	– Расстояние редактирования графа (*graph edit distance*)
msd	– Процедуры сравнения среднего графа с удаленным одиночным
$c(e)$	– Стоимость операции e, переводящей один граф в другой
$c(s)$	– Суммарная стоимость операций, переводящих один граф в другой
$\gamma(u_i)$	– Число повторений вершины u_i в последовательности графов
MCS	– Максимально общий подграф (*maximal common subgraph*)

Анализ методов и инструментов отладки распределённых информационных систем

В архитектуре РИС сохраняется высокая вероятность возникновения ошибок, обусловленных особенностями сетевого взаимодействия, отказами отдельных узлов и сложностью процессов синхронизации и согласования данных. В связи с этим традиционные подходы отладки оказываются малоэффективными, что обусловливает необходимость применения наряду с классическими специализированных методов, которые можно разделить на несколько групп [3]:

– *журналирование и мониторинг событий*. В основе этой группы методов лежит сбор журналов и метрик работы программных компонентов РИС. Для последующей агрегации данных применяются инструменты

централизованного сбора и хранения. Визуализация при этом реализуется, как правило, в рамках подсистемы мониторинга. Данный подход характеризуется полнотой и информативностью данных, однако сопровождается большими объёмами информации и требует сложной инфраструктуры;

– *распределённая трассировка*. Данный метод является наиболее распространённым при диагностике РИС, основанных на стандартах OpenTracing [4] и OpenTelemetry [5]. Метод позволяет отслеживать путь (трассу) запроса между сервисами, формируя последовательность событий с уникальными идентификаторами трасс, характеризующих отдельные пользовательские запросы. В отличие от журналирования, трассировка обеспечивает целостное представление о прохождении запроса сквозь систему, однако требует модификации кода и интеграции с библиотеками. Наибольшее распространение получили инструменты: ShiViz [6] – анализ журналов с преобразованием их в диаграммы причинно-следственных связей; Jaeger [7] – поддержка высоконагруженных систем, построение временных диаграмм запросов; Zipkin [8] – масштабируемое решение с хранением данных в ElasticSearch [9] и Dapper [10];
– *удалённая отладка*. Данный метод используется за счет подключения отладчика к удалённым узлам РИС. Как и предыдущие два метода, данный подход позволяет оценить работоспособность системы в реальном масштабе времени. Он позволяет выявлять ошибки на программном уровне, но требует дополнительных усилий связанных с интеграцией средств отладки в структуру узлов. Данный метод получил распространение именно при отладке логических ошибок РИС. Использование метода для отслеживания ошибок вследствие деградации производительности системы затруднено по причине значительной нагрузки, создаваемой распределенными средствами отладки;
– *интеграционные и сквозные тесты*. Данные методы ориентированы на проверку взаимодействия между компонентами и всего цикла обработки запросов. Применение интеграционных тестов позволяет выявить ошибки, связанные с форматом передаваемых данных, некорректной обработкой запросов и ответов, а также совместимости при использовании множества различных протоколов обмена. Сквозные тесты ориентированы на проверку полного цикла работы системы на множестве узлов – от начала обработки запроса до получения конечного ответа;
– *моделирование сбоев и отказов*. Данная группа методов предполагает создание искусственных нестабильных условий: задержек в передаче данных, потерь и переупорядочивания пакетов и джиттера, ухудшение качества обслуживания (имитация ухудшения работы конкретных компонентов системы для проверки возможности поддержания системой общего уровня производительности).

Каждый из рассмотренных методов обладает рядом достоинств и недостатков, таблица 2.

Таблица 2 – Преимущества и недостатки методов отладки распределённых информационных систем

Метод	Преимущества	Недостатки
Журналирование и мониторинг	Реальные данные о состоянии системы; высокая информативность; возможность настройки отображения статистик	Большие объёмы информации; необходимость реализации сложной инфраструктуры сбора и обработки данных
Распределённая трассировка	Наглядность модели взаимодействия компонентов; возможность локализации узких мест и задержек	Требует модификации программного кода компонентов системы; увеличивает активную нагрузку на систему
Удалённая отладка	Возможность поиска и отладки логических ошибок; доступ к актуальному состоянию узлов в реальном времени	Трудоёмкость выполнения; риск влияния на компоненты системы; необходимость ручной настройки
Интеграционные и сквозные тесты	Выявление ошибок на стыках модулей, баз данных, внешних сервисов; возможность частичной автоматизации	Сложность разработки и поддержки; необходимость учёта асинхронности; ограниченность масштабируемости при увеличении количества узлов РИС
Моделирование сбоев и отказов	Возможность выявления скрытых дефектов; оценка устойчивости системы при сбоях	Высокая стоимость настройки тестового окружения (окружение, должно быть максимально приближено к условиям реальной эксплуатации); риск негативного влияния на систему; трудоёмкость при организации сценариев (сеть, узлы, внешние сервисы).

Для полноты картины проведем также сравнительный анализ наиболее распространенных инструментов распределенной отладки и анализа по следующим ключевым критериям:

– возможность анализа объединённых журналов с нескольких узлов;
– локализация переходных состояний РИС;
– оказание активной нагрузки на РИС;
– поддержка визуализации временных диаграмм;
– визуализация трасс событий (последовательностей программных вызовов) в виде графов взаимодействия компонентов;
– фильтрация логов по заданным критериям.

Результаты сравнения представлены в таблице 3.

На основании проведенного анализа можно заключить, что перспективный инструментарий анализа РИС должен иметь следующие особенности:

– возможность трассировки в процессе взаимодействия программных компонентов РИС;
– децентрализованная (локальная) журнализация состояния компонентов с возможностью создания объединенного журнала;

– отсутствие существенной дополнительной нагрузки на РИС, вызванной работой инструментария (агенты, отладчики и пр.);
– возможность поиска и локализации периодов аномального состояния с визуализацией состояния РИС.

Таблица 3 – Сравнительная характеристика инструментов

Критерий/Инструмент	ShiViz	Jaeger	Zipkin	Dapper
Возможность анализа объединенного журнала с нескольких узлов	Да	Да	Да	Да
Возможность локализации переходных состояний процессов системы между нормальным состоянием и сбоем (интервал начала и окончания сбоя)	Да	Нет	Нет	Нет
Возможность использования без активной нагрузки на систему	Да	Нет	Нет	Нет
Возможность визуализации временных диаграмм	Нет	Да	Да	Да
Возможность визуализации трасс событий в виде графов взаимодействия компонентов	Да	Нет	Нет	Да
Возможность фильтрации журналов по заданным критериям	Да	Да	Да	Нет

При анализе РИС в качестве исходных данных следует рассматривать не только отдельные показатели загрузки процессора или потребления памяти, как это характерно для традиционных систем мониторинга. Указанные метрики необходимо соотносить с фиксируемыми в журналах последовательностями программных вызовов. При этом задача отображения трасс журналов с целью выявления интервалов аномального состояния может быть решена посредством применения графо-аналитических методов как это показано в [11]. Действительно, графы состояния, формируемые совокупностью трасс (последовательностей вызовов) за определённый промежуток времени, различаются между собой не только по параметрам, но и по структуре. Использование в качестве меры различия последовательных состояний РИС расширенной метрики расстояния редактирования позволяет учесть эту особенность. Далее рассмотрим данный подход более подробно.

Графо-аналитический метод идентификации аномального состояния распределенных информационных систем

Схема предложенного метода приведена на рис. 1. Исходными данными являются необработанные журналы событий, включающие как информативные записи о программных вызовах компонентов РИС, так и служебные сообщения (старт компонента, ошибки и исключения и пр.). В связи с этим первый этап направлен на предварительную обработку и объединение журналов отдельных компонентов РИС в унифицированный единый журнал. Каждая запись унифицированного журнала содержат следующие атрибуты:

– идентификатор программного потока(thread) и распределенной транзакции;
– выполняемое действие;
– временная метка;
– этап выполнения действия (начало, завершение или сбой).

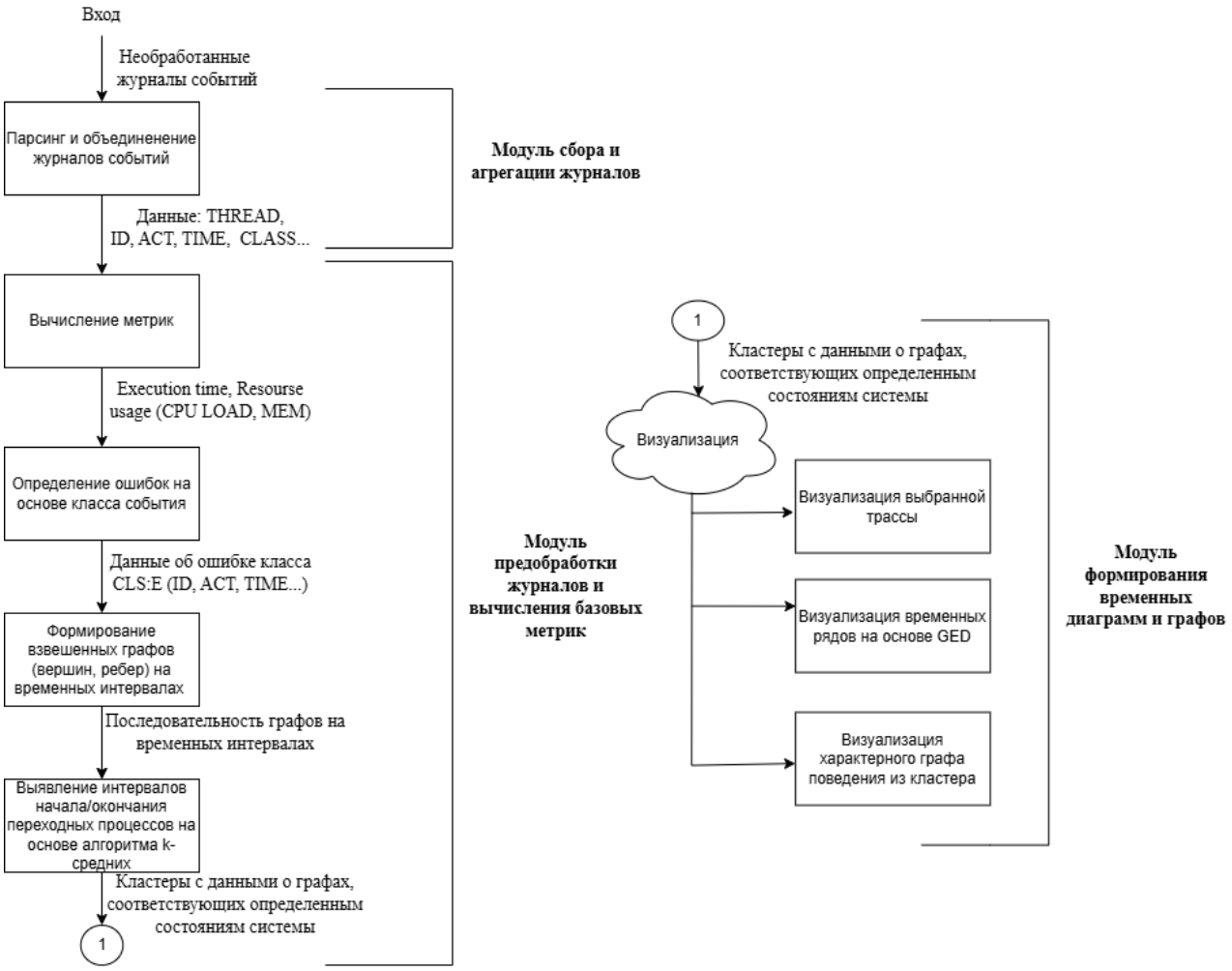

Рис. 1. Схема графо-аналитического метода идентификации аномального состояния распределенной информационной системы

На следующем этапе осуществляется вычисление метрик, отражающих характеристики функционирования системы. В качестве метрик были использованы следующие показатели:

– время выполнения действий, соответствующих вызову функций компонентами распределенной информационной системы;
– временной интервал между последовательными действиями;
– коэффициент использования вычислительных ресурсов на момент фиксации события в журнале (процессорное время, оперативная и дисковая память).

Следующим, ключевым этапом метода является построение временных последовательностей усредненных графов поведения РИС, формируемых для каждого временного интервала, на который разбивается объединенный журнал

состояния системы. В данной модели события интерпретируются как вершины, а причинно-следственные связи между ними – как рёбра графа.

Для анализа динамики изменений применяется мера графового расстояния редактирования (Graph Edit Distance, GED) [9], позволяющая сопоставлять последовательности графов и выявлять структурные и динамические отклонения. На интуитивном уровне, расстояние редактирования – минимальное число операций вставки, удаления и замены вершин и ребер, необходимое для того, чтобы сделать два графа изоморфными (эквивалентными). Общая формула графового расстояния имеет следующий вид:

$$d(G,G')=1-\frac{|MCS(G,G')|}{\max\{|G|,|G'|\}},$$

где $MCS(G,G')$ – максимальный общий граф для графов G_1 и G_2, $|G|$– число вершин (или ребер в графе), $\max\{|G|,|G'|\}$ – максимальное количество вершин/ребер, попавших в выбранные графы.

Введенная метрика расстояния редактирования между графами позволяет осуществить их разбиение на классы, соответствующие различным состояниям РИС (нормальная работа, предотказный и аварийный режим). Для этого на следующем этапе применяется кластеризация с использованием метода *k*-средних. Усреднённые (медианные) графы, соответствующие центрам кластеров, формируют эталонные модели состояния системы. Сопоставление текущих графов с этими эталонными структурами позволяет выявлять интервалы начала и завершения переходных процессов.

Результаты анализа могут быть представлены в различных формах, включающих диаграммы графов, построенных на основе выбранных трасс событий (последовательностей программных вызовов), временные ряды значений графового расстояния, а также усреднённые графовые модели, характерные для различных режимов функционирования РИС.

Рассмотрим более подробно алгоритм кластеризации графов поведения РИС на различных временных интервалах. Блок-схема данного алгоритма представлена на рис. 2.

Шаг 1. На вход алгоритма поступает множество ориентированных графов вида $\{G_0, G_1, \ldots, G_n\}$, где каждый граф $G_i = (V_i, E_i, \alpha_i, \beta_i)$ соответствует временному интервалу $T_i = [t_i, t_{i+1}]$. Множество вершин графа V_i формируется действиями, выполняемыми программными компонентами РИС, а ребра $E_i \subseteq V_i \times V_i$ соответствуют порядку выполнения действий в последовательности событий журнала. Функций весов вершин α_i: $V_i \rightarrow R$ и ребер β_i: $E_i \rightarrow R$ сопоставляет соответственно вершинам и ребрам набор характеристик, включающий показатели потребления ресурсов процессора и памяти, а также время выполнения операций.

Шаг 2. Множество графов подвергается кластеризации и разделяется на *k* подмножеств $\{C_1, C_2, \ldots, C_k\}$, где каждый кластер C_j соответствует устойчивому состоянию системы S_j. Число кластеров *k* задаётся априори и является особенностью метода *k*-средних. В рассматриваемом случае кластеры соотносятся с

состояниями нормального, предотказного и аварийного функционирования системы, обозначаемыми как S_1, S_2, …, S_i.

Рис. 2. Блок-схема алгоритма кластеризации графов поведения РИС в предлагаемом методе

Шаг 3. Для каждого кластера C_j определяется его центр, представленный в виде среднего (усреднённого) графа, вычисляемого по формуле:

$$\mu_j^{(0)} = MCS\left(G_1^j,...,G_n^j\right),$$

где *MCS* (Maximum Common Subgraph) – операция выделения общего подграфа или медианного графа, минимизирующего сумму расстояний до графов в C_j, представленная выражением:

$$\mu_j = \arg\min_{G\in C_j} \sum_{G\in C_j} d(G,G'),$$

где $d(G,G')$ – мера графового расстояния.

При этом медианный граф для графового расстояния $d(G,G')$ в общем случае можно вычислить следующим образом.

Пусть задано множество графов $g=\{G_0,G_1,...,G_n\}$, тогда средний граф задается как $\widehat{G}=\left(\widehat{V},\widehat{E},\widehat{\beta},\widehat{\delta}\right)$, где $\widehat{V}$ – множество средних вершин, $\widehat{E}$ – множество

средних ребер, $\hat{\beta}$ – множество средних весов ребер (временная разница между событиями), $\hat{\delta}$ – множество средних весов вершин (принимает значения загрузки процессора и памяти), и определяется как:

$$\hat{V} = \left\{ u \mid u \in V, \gamma(u) > \frac{n}{2} \right\},$$

$$\hat{E} = \left\{ (u,v) \mid u,v \in \hat{V} \right\},$$

$$\hat{\beta}(u,v) = \text{median}\left\{ \beta_i(u,v) \mid i = 1,...,n \right\},$$

$$\hat{\delta}(u) = \text{median}\left\{ \delta_i(u) \mid i = 1,...,n \right\},$$

где u – вершины, (u,v) – ребра, $\gamma(u)$ – число вхождений вершины u в графах G_i, median {} – функция вычисления центрального значения множества.

Шаг 4. Каждый граф G_i сопоставляется ближайшему кластеру на основе рассмотренной метрики расстояния между графами $d(G_i, \mu_j)$. Таким образом, граф присваивается кластеру C_i, для которого расстояние минимально, то есть соответствует формуле:

$$j = \arg\min_{1 \le j \le k} (G_i, \mu_i).$$

Шаг 5. После распределения состава кластеров для каждого C_j вычисляется новый центр:

$$\mu_j^{t+1} = \arg\min_{G \in C_j} \sum_{G' \in C_j} d(G, G').$$

Шаг 6. Производится итеративный повтор шагов 3-5 до выполнения условия сходимости (центры кластеров не изменяются), заданного следующим выражением:

$$\forall j \in \{1,...,k\} \Rightarrow \mu_j^{(t+1)} = \mu_j^{(t)}.$$

При этом процесс может завершиться и по причине достижения заданного количества итераций $T_{\max}$.

Шаг 7. По завершении кластеризации получается разбиение множества графов g на кластеры $\{C_1,...,C_k\}$, отражающие состояние системы (нормальное, предотказное, аварийное) на интервалах времени.

Формируемый для каждого временного интервала граф поведения РИС может быть соотнесен с одним из кластеров и визуализирован, например, при помощи точечной диаграммы. Это будет наглядно проиллюстрировано в следующем разделе с временными периодами, соответствующих различным режимам работы РИС на точечной диаграмме.

Экспериментальное исследование

Экспериментальное исследование, выполненное на макете фрагмента корпоративной распределённой информационной системы, было направлено на апробацию предложенного метода при выявлении трудновоспроизводимых ошибок. В качестве тестовой РИС была использована корпоративная система электронного документооборота, развернутая на нескольких узлах, в которой

наблюдался сбой в цикле репликации после длительного времени работы (порядка 3–4 ч).

На всех узлах была активирована система журналирования, фиксировавшая события, возникающие в процессе функционирования подсистемы репликации данных. Для этого были определены точки логирования, обеспечивающие регистрацию как технических событий межузлового взаимодействия, так и внутренних операций, связанных с обработкой сессий, транзакций и формированием пакетов репликации.

Каждое событие, зафиксированное в журналах, было отнесено к определённому этапу работы механизма репликации. Для вторичных серверов регистрировались операции создания (или восстановления) сессии репликации (CREATE_SESSION_BASED_OLD, CREATE_SESSION_SEC), открытия и завершения транзакций (OPEN/COMMIT_SESSION_TRANS_SEC), формирования и передачи пакетов репликации на первичный сервер (BEG/COMMIT TRANS_PREP_SEC, JSON_PREP_SEC, SEC_SEND_PRIM,), а также обработки и развёртывания ответных сообщений от него (RCV_SEC_FROM_PRIM, DESER_SEC_FROM_PRIM, DEPLOY_TRANS_CREATE и т. д.). Для первичного узла фиксировались события приёма пакетов, их развёртывания (UNFINISHED_PRIM, BEG_RECV_PRIM, BEG_TRANS_PRIM и т.д.), а также формирования и передачи ответных пакетов вторичным серверам (BEGIN_TRANS_PRIM_SEC, REMOVE_PACK_PRIM, RESP_PRIM_PREPARE, COMMIT_TRANS_PRIM_SEC).

Цель эксперимента – проверка отказоустойчивости распределенной информационной системы и проверка возможности поиска аварийных ситуаций предложенным методом.

Эксперимент проводился в контролируемой среде, развернутой на пяти логических узлах: UC1, UC2, UC3, UC4 и UC5. При этом узел UC1 выполнял функции первичного сервера, тогда как узлы UC2 – UC5 выступали в роли вторичных (подчинённых). Анализ состояния системы осуществлялся в двух режимах работы:

– *режим пиковой нагрузки.* Был сформирован тестовый план, генерировавший параллельные запросы к главному серверу UC1 с частотой один запрос в секунду в течение пяти минут. После завершения этапа нагрузки сбор данных продолжался ещё в течение 1 ч 30 мин для контроля свойств релаксации распределенной информационной системы после нагрузки;
– *режим штатного функционирования.* В рамках тестового плана выполнялся один запрос каждые десять секунд в течение одного часа. В завершение эксперимента один из вторичных серверов был досрочно остановлен.

Результаты анализа, включающие временные ряды на основе графового расстояния (GED), граф состояния системы из кластера, соответствующего нормальному состоянию, а также граф взаимодействия компонентов выбранной трассы, представлены на рис. 3–5.

На рис. 3 представлено отображение в виде временного ряда реакции РИС на пиковую нагрузку. Исходя из рисунка, основная нагрузка выпала на временной интервал с 14:11 по 14:20. Красной линией обозначена функция расстояния (GED) между двумя соседними графами, синей – графовое расстояние, полученное в результате сравнения со средним графом, вычисленным по скользящему окну (процедура *mma* [9, 11]). Ось абсцисс соответствует временным интервалам, а ось ординат – значениям меры графового расстояния (GED) с учетом нормализованных весов вершин (коэффициент загрузки процессора, оперативной и дисковой памяти) и весов ребер (временной интервал между действиями). Отсутствие в начальный период наблюдения динамики синего графика объясняется предварительным подсчетом скользящего окна на основании заданных 15 графов, используемых в процедуре сравнения среднего графа с последующим средним *mma* [9, 11]. Отображенные в нижней части точки соответствуют соотнесению графа временного интервала с одним из кластеров, отражающего состояние системы. Нормальное, предотказное и аварийное состояния явно определены на графике: синему цвету соответствует нормальное, красному – аварийное, зеленому – предотказное.

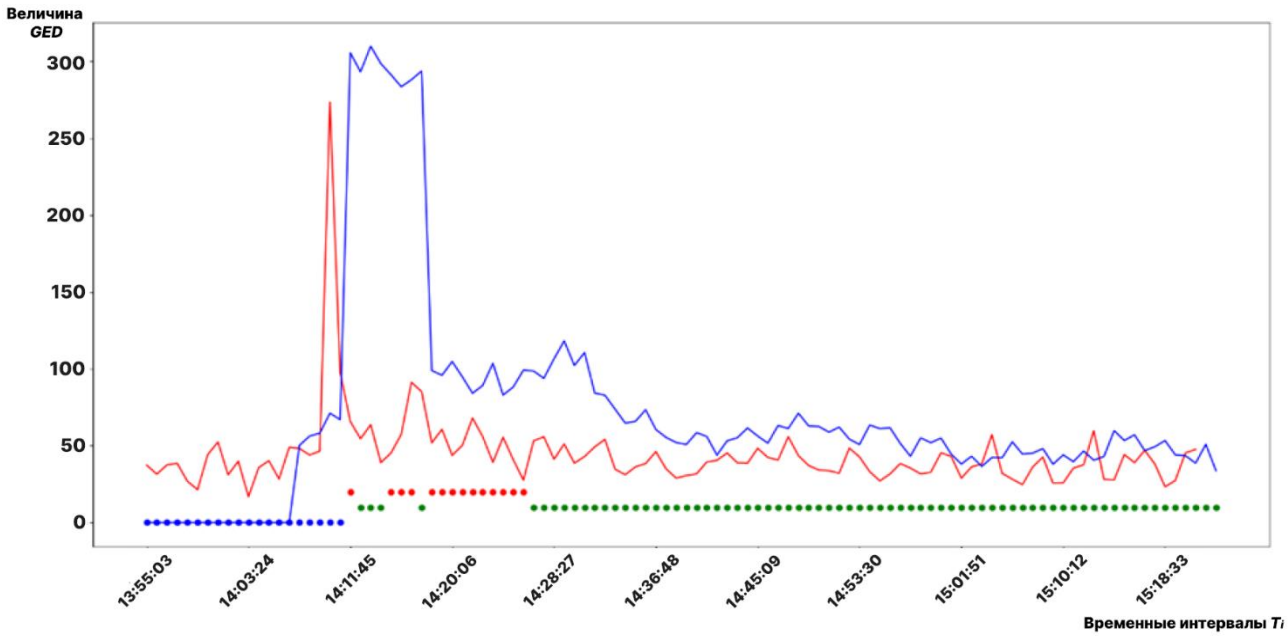

Рис. 3. Диаграмма временных рядов на основании GED

На рис. 4 представлен граф поведения РИС, соответствующий центру кластера нормального состояния. Здесь каждое событие, обозначенное узлом (вершиной в графе) имеет собственную цветовую индикацию от UC1 до UC5. Из рисунка видно, что число операций на главном сервере UC1 (обозначен на графе синим цветом) значительно меньше, чем на вторичных (события с узлов: UC2 – зеленый цвет, UC3 – желтый, UC4 – красный, UC5 – оранжевый), что объясняется особенностями отображаемых этапов репликации. Значения на ребрах соответствуют временному интервалу между событиями. Вес дублируется на схеме толщиной ребра.

DOI: 10.24412/2410-9916-2025-4-179-199
URL: https://sccs.intelgr.com/archive/2025-04/08-Buravlev.pdf

№4. 2025
ISSN 2410-9916

Рис. 4. Граф состояния РИС (нормальное состояние системы)

Из рис. 4 следует, что большинство толстых ребер приходятся на события создания сессии репликации, а также прием и развертывание пакетов, что может свидетельствовать о наличии «узкого места» при наличии сетевых задержек или при повышенной загрузке системы управления базами данных узла РИС.

Дополнительно у каждой вершины графа отображаются значения весов коэффициента загрузки процессора (CPU) и памяти (MEM), соответствующие нормальному состоянию системы. К примеру, на представленной диаграмме рис. 4 не зафиксировано событий, превышающих загрузку процессора более чем на 15 %, а максимальная временная разница между исполнениями событий составляет около 5 с.

На рис. 5 представлена визуализация выбранной трассы событий в виде графа взаимодействия компонентов. Например, показанная трасса, принадлежащая интервалу с 14:13:25 по 14:14:15, соответствует согласно рис. 3 пиковой нагрузке. На графике явным образом выделено красной цветовой рамкой событие, завершившееся с ошибкой. При этом коэффициент загрузки процессора соответствуют 100 %, что точно отражает пиковую нагрузку при развертывании пакета на первичном сервере.

Полученные граф взаимодействия компонентов выбранной трассы событий из наиболее загруженного интервала и диаграммы временных рядов на основании графового расстояния (GED), представлены на рис. 6 и 7.

Анализ рис. 6 показывает, что в выбранную для построения графа взаимодействия компонентов трассу событий, относящуюся к временному интервалу 18:05:45–18:06:22, вошли исключительно события, зафиксированные на одном из подчинённых узлов. Данный факт свидетельствует о плановом выходе из строя первичного сервера UC1.

На рис. 7 в виде точек в нижней части графика показано соотнесение состояния с одним из кластеров. Например, с самого начала интервалов от 16:52:56 наблюдается равномерная нагрузка с небольшим скачком в изменении значений графового расстояния от 17:20 до 17:40. В данном случае функции меры расстояния на основе сравнения соседних графов (красная кривая) и с помощью сравнения каждого текущего графа с заданным скользящим окном (синяя кривая) примерно одинаково отображают состояние системы, поскольку в конце графика наблюдается значительный скачок значений, преуменьшающий уровень значимости значений, попавших в предыдущие интервалы.

Заключение

В настоящей работе предложен и апробирован графо-аналитический метод идентификации аномального состояния распределенных информационных систем, основанный на построении усреднённых моделей поведения с использованием графового представления процессов. Методологической основой такого подхода является интерпретация событий и их взаимосвязей в виде ориентированных графов с весовыми признаками, что обеспечивает возможность не только параметрического, но и структурного анализа функционирования РИС.

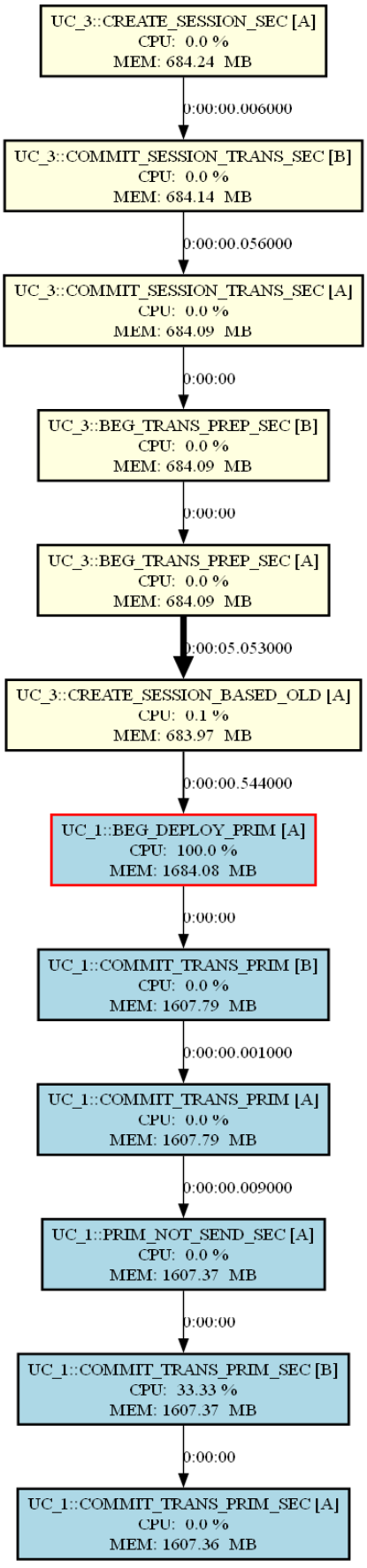

Рис. 5. Граф взаимодействия компонентов выбранной трассы событий из наиболее загруженного интервала

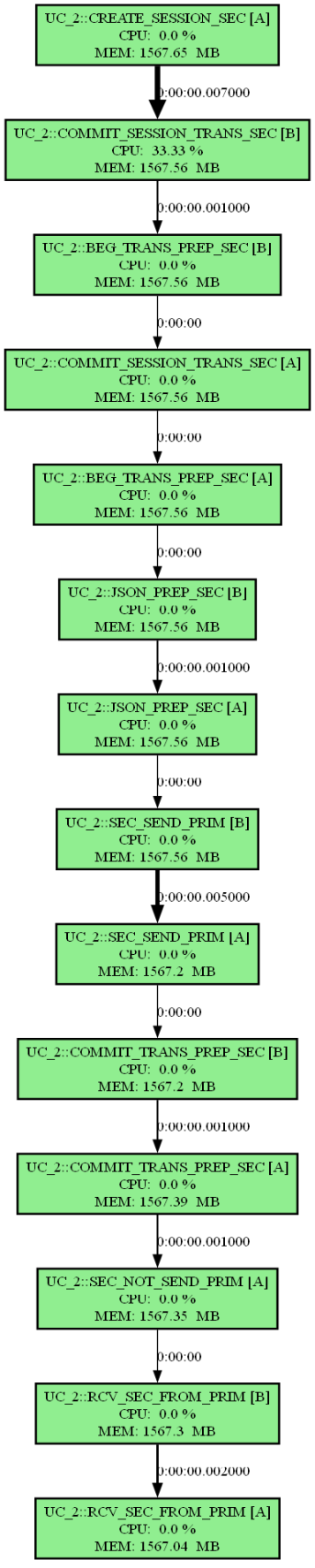

Рис. 6. Граф взаимодействия компонентов (штатный режим работы)

№4. 2025
ISSN 2410-9916

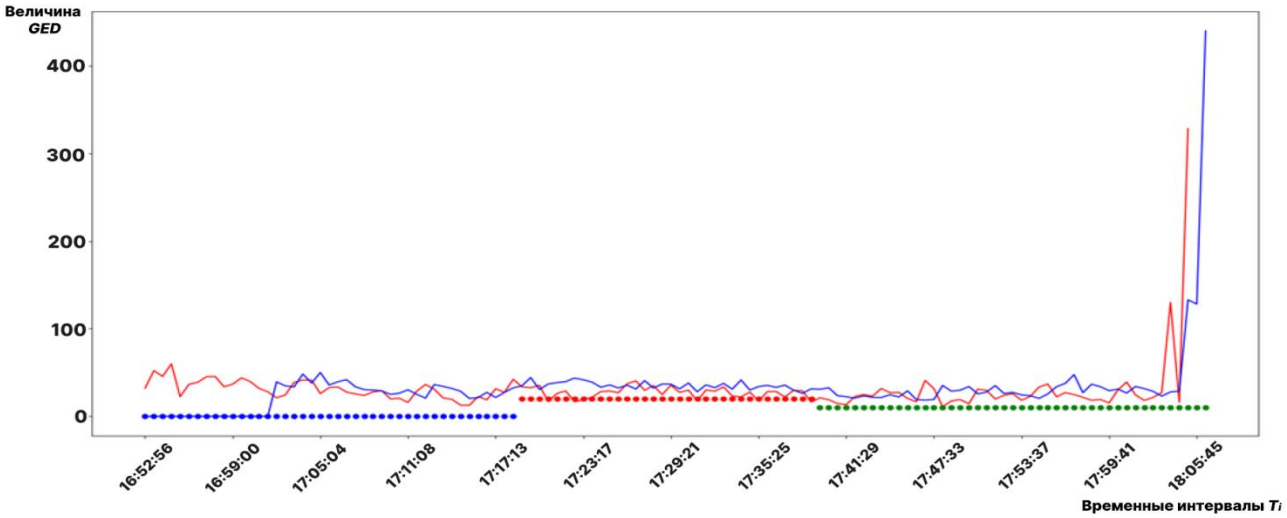

Рис. 7. Диаграмма временных рядов на основании GED
(штатный режим работы)

Разработанный прототип программного комплекса подтвердил практическую реализуемость предложенного подхода. В частности, в его рамках обеспечена интеграция ключевых этапов обработки данных:

- предварительная очистка и унификация журналов, поступающих с различных узлов;
- преобразование их в единый формат;
- вычисление метрик времени выполнения и потребления ресурсов;
- фиксация ошибок;
- построение графов взаимодействия компонентов;
- выделение переходных процессов между нормальными и аномальными состояниями системы.

Проведённое экспериментальное исследование на натурном макете продемонстрировало, что использование метрики графового расстояния (GED) позволяет выявлять изменения, как в параметрическом, так и в структурном состоянии распределённой информационной системы. Это обеспечивает возможность фиксации интервалов перехода от нормального функционирования к состояниям перегрузки, предотказного режима и последующих сбоев. Полученные визуализации в форме графов трасс событий, усреднённых эталонных моделей и временных рядов подтвердили диагностическую ценность метода и его применимость для анализа сложных РИС.

В отличие от традиционных средств логирования, трассировки и мониторинга, предложенный метод обеспечивает комплексное объединение информации с различных узлов без необходимости модификации прикладного кода. Это существенно расширяет его потенциал для промышленного применения, так как позволяет выявлять трудновоспроизводимые ошибки и локализовать переходные процессы, недоступные для классических средств диагностики.

DOI: 10.24412/2410-9916-2025-4-179-199
URL: https://sccs.intelgr.com/archive/2025-04/08-Buravlev.pdf

Перспективными направлениями дальнейших исследований являются:

- расширение выборки журналов и проведение апробации метода на гетерогенных распределённых информационных системах с высокой интенсивностью трафика;
- исследование масштабируемости метода при росте числа узлов и объёмов данных;
- интеграция подхода с существующими инструментами мониторинга и средствами автоматизации диагностики;
- разработка методов прогнозирования сбоев на основе кластеризации графов и анализа динамики переходных процессов;
- применение элементов машинного обучения для повышения точности и автоматизации выделения аномальных режимов.

Таким образом, результаты работы подтверждают актуальность и эффективность предложенного графо-аналитического метода идентификации аномального состояния распределенных информационных систем, демонстрируют его преимущества перед существующими средствами диагностики и открывают возможности для дальнейшего развития в направлении интеллектуального мониторинга и проактивного управления отказоустойчивостью распределенных информационных систем.

Литература

1. ГОСТ 34.321–96. Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными. – М.: Госстандарт России, 1996. – 28 с.

2. Котенко И. В., Саенко И. Б., Лаута О. С., Крибель А. М. Методика обнаружения аномалий и кибератак на основе интеграции методов фрактального анализа и машинного обучения // Информационная безопасность. 2021. № 6. С. 9–21. DOI: 10.15622/ia.21.6.9.

3. Maximilian Michels BLOG. The Art of Debugging Distributed Systems. [Электронный ресурс]. – URL: https://maximilianmichels.com/2020/debugging-distributed-systems/ (дата обращения: 12.10.2025).

4. Документация по Elasticsearch. [Электронный ресурс]. – URL: https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html (дата обращения: 12.10.2025).

5. Документация по Jaeger. [Электронный ресурс]. – URL: https://www.jaegertracing.io/docs/1.18/ (дата обращения: 12.10.2025).

6. Документация по Zipkin [Электронный ресурс]. – URL: https://zipkin.io/ (дата обращения: 12.10.2025).

7. Документация по ShiViz [Электронный ресурс]. – URL: https://bestchai.bitbucket.io/shiviz/ (дата обращения: 12.10.2025).

8. Обзор стандарта OpenTracing [Электронный ресурс]. – URL: https://opentracing.io/docs/overview/ (дата обращения: 12.10.2025).

9. Обзор стандарта OpenTelemetry [Электронный ресурс]. – URL: https://opentelemetry.io/docs/migration/opentracing/ (дата обращения: 12.10.2025).

10. Документация по Dapper [Электронный ресурс]. – URL: https://www.learndapper.com/ (дата обращения: 12.10.2025).

11. Будко Н. П., Васильев Н. В. Обзор графоаналитических подходов к мониторингу информационно-телекоммуникационных сетей и их применение для выявления аномальных состояний // Системы управления, связи и безопасности. 2021. № 6. С. 53-75. DOI: 10.24412/2410-9916-2021-6-53-75.

References

1. Russian State Standard 34.321–96. Information Technology. Database Standards System. Reference Model for Data Management. Moscow, Gosstandart Rossii Publ., 1996. 28 p. (in Russian).

2. Kotenko I. V., Saenko I. B., Lauta O. S., Kribel A. M. Methodology for detecting anomalies and cyberattacks based on the integration of fractal analysis and machine learning methods. *Information Security*, 2021, no. 6, pp. 9–21 (in Russian). DOI: 10.15622/ia.21.6.9.

3. Maximilian Michels BLOG. The Art of Debugging Distributed Systems. [Electronic resource]. Available at: https://maximilianmichels.com/2020/debugging-distributed-systems/ (accessed 12 October 2025).

4. Elasticsearch documentation. Available at: https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html (accessed 12 October 2025).

5. Documentation on Jaeger. Available at: https://www.jaegertracing.io/docs/1.18/ (accessed 12 October 2025).

6. Zipkin documentation. Available at: https://zipkin.io/ (accessed 12 October 2025).

7. ShiViz documentation. Available at: https://bestchai.bitbucket.io/shiviz/ (accessed 12 October 2025).

8. Overview of the Open Tracing standard. Available at: https://opentracing.io/docs/overview/ (accessed 12 October 2025).

9. Overview of the Open Telemetry standard. Available at: https://opentelemetry.io/docs/migration/opentracing/ (accessed 12 October 2025).

10. Dapper documentation. Available at: https://www.learndapper.com/ (accessed 12 October 2025).

11. Budko N. P., Vasiliev N. V. Review of graph-analytical approaches to monitoring of information and telecommunication networks and their application to identify abnormal states. *Systems of Control, Communication and Security*, 2021, no. 6, pp. 53-75 (in Russian). DOI: 10.24412/2410-9916-2021-6-53-75.

Статья поступила 19 октября 2025 г.

Информация об авторах

Буравлев Андрей Сергеевич – аспирант. Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина). Область научных интересов: методы обработки естественного языка, анализ бизнес-процессов. E-mail: asburavlev@stud.etu.ru

Адрес: 197022, Санкт-Петербург, ул. Профессора Попова, д. 5, лит. Ф.

Васильев Николай Владимирович – кандидат технических наук, доцент. Доцент кафедры информационных систем. Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина). Область научных интересов: методы обработки естественного языка, анализ бизнес-процессов. E-mail: gandvik1984@gmail.com

Адрес: 197022, Санкт-Петербург, ул. Профессора Попова, д. 5, лит. Ф.

Демидова Дарья Евгеньевна – студентка. Национальный исследовательский университет ИТМО. Область научных интересов: методы обработки естественного языка, анализ бизнес-процессов. E-mail: daryademiddovaa@yandex.ru

Адрес: 197101, Санкт-Петербург, Кронверкский пр., д. 49, лит. А.

Будко Никита Павлович – кандидат технических наук. Независимый специалист. Область научных интересов: мониторинг информационных ресурсов; сбор и обработка информации. E–mail: budko62@mail.ru

Адрес: 194064, г. Санкт-Петербург, ул. Бутлерова, 9, корп. 3, кв. 252.

Graphical-analytical method for identifying abnormal states in distributed information systems

A. S. Buravlev, N. V. Vasiliev, D. E. Demidova, N. P. Budko

Problem statement: *Modern distributed information systems are highly complex, scalable, and require fault tolerance, making them difficult to diagnose. Existing methods for identifying the abnormal state of such systems are used to establish the occurrence of an incident and logs of the functioning of individual nodes. However, the underlying methodological basis does not fully localize the failure in the structure of a distributed information system and does not allow for an analysis of the dynamics of its development. In this paper, we propose a method for identifying abnormal states using a grapho-analytical approach for modeling and analyzing their development dynamics in distributed information systems. To build models of the state of a distributed information system over time ("snapshots"), a combined log is used, collected from its components (nodes) and describing the sequence of combined program calls in the process of computing and communication. The method includes the stage of cleaning and unifying logs of components of a distributed information system, the stage of extracting metrics of computing resource usage from log records, the stage of modeling the behavior of a distributed information system in the form of sequences of weighted graphs, and the actual stage of identifying abnormal states based on clustering graphs constructed at the previous stage using the k-means method as the editing distance increases. Each cluster corresponds to one of the states of the distributed information system. As a result of clustering, the average models of the "typical" behavior of the system in each of the states are also determined. Correlating each of the behavior graphs with one of the clusters allows us to identify the intervals of abnormal activity and the moments of transitions between normal and emergency states. Further retrospective analysis of the weighted graphs in the selected sections allows us to deepen our understanding of the processes taking place.* ***Methods used:*** *graph theory methods; data analysis methods; process extraction technologies; system analysis methods.* ***The novelty of the work:*** *the novelty of the research is determined by the use of grapho-analytical methods to identify the*

abnormal state, their refinement for use in the analysis of logs of distributed information systems. ***Result:*** *the method described in the paper is implemented as a prototype of a software package that provides visualization of the dynamics of behavior of a distributed information system in the form of graphs, analysis of changes in their structure over time (editing distance), as well as identification of moments of abnormal behavior in time.* ***Practical significance:*** *the results showed that the method makes it possible to identify difficult-to-reproduce errors and monitor the development of node congestion, failures and synchronization violations. Unlike existing solutions, the proposed approach provides a comprehensive analysis of combined logs from multiple nodes without modifying the application code, which confirms its applicability to industrial distributed information systems.*

Keywords: *average graph, distributed information systems, event log, graph editing distance, graph clustering, monitoring, tracing.*

About the Authors

Buravlev Andrey Sergeevich – Postgraduate. St. Petersburg State Electrotechnical University named after V. I. Ulyanov (Lenin). Field of research: natural language processing methods, business process analysis. E-mail: asburavlev@stud.etu.ru

Address: 197022, Russia, St. Petersburg, Professora Popova street, house 5, letter F.

Vasiliev Nikolay Vladimirovich – Ph.D. of Engineering Sciences. Associate Professor at the Department of Information Systems. St. Petersburg State Electrotechnical University named after V. I. Ulyanov (Lenin). Field of research: natural language processing methods, business process analysis. E-mail: gandvik1984@gmail.com

Address: 197022, Russia, St. Petersburg, Professora Popova street, house 5, letter F.

Demidova Daria Evgenevna – Student. ITMO National Research University. Field of research: natural language processing methods, business process analysis. E-mail: daryademiddovaa@yandex.ru

Address: 197101, Russia, St. Petersburg, Kronverksky ave., 49, lit. A.

Budko Nikita Pavlovich – Ph.D. of Engineering Sciences. An independent specialist. Field of research: monitoring of information resources; collection and processing of information. E–mail: budko62@mail.ru

Address: 194064, Russia, St. Petersburg, Butlerova St., 9, building 3, sq. 252.