

СТОХАСТИЧЕСКАЯ МОДЕЛЬ СВЯЗНОСТИ МИКРОСЕРВИСНЫХ СИСТЕМ ДЛЯ ОЦЕНКИ ДОСТУПНОСТИ

А.А. Красновский^{1,2}

¹Университет Иннополис, г. Иннополис

²MB3R Lab, г. Иннополис

e-mail: a.krasnovskii@innopolis.university

Аннотация

Предлагается единая стохастическая модель связности микросервисной системы для оценки доступности пользовательских запросов при независимых отказах с остановкой. Модель задается графом зависимостей, вектором репликации, вероятностной мерой на пространстве состояний и предикатом успешности. Доступность оценивается методом Монте-Карло. Валидация на стендах DeathStarBench и OpenTelemetry Demo показывает согласие с экспериментом и уточняет роль асинхронных связей для немедленного HTTP-ответа.

Ключевые слова: микросервисы, доступность, вероятностная графовая модель, распределенная трассировка, Монте-Карло.

Введение

Количественная оценка доступности микросервисной системы затруднена большим числом межсервисных зависимостей, репликацией и быстрым ростом пространства состояний отказов. Натурные эксперименты по внесению отказов дают прямую проверку устойчивости, но их трудно выполнять широко и регулярно [1]. Поэтому интерес представляет модель, которая сохраняет существенные для доступности свойства архитектуры и при этом допускает быстрый численный расчет.

В работе предлагается единая стохастическая модель связности микросервисной системы. Ее структура восстанавливается по данным распределенной трассировки, а корректность модели проверяется на стендах DeathStarBench Social Network [2] и OpenTelemetry Demo [3].

1. Стохастическая модель связности

Микросервисная система описывается кортежем

$$M = \langle G, R, P, \Phi \rangle, \quad G = (V, E, \tau). \quad (1)$$

Здесь V – множество типов сервисов, $E \subseteq V \times V$ – множество зависимостей, τ определяет тип каждой зависимости (синхронная или асинхронная связь), $R: V \rightarrow \mathbb{N}$ задает число реплик, P – вероятностную меру на пространстве состояний Ω , Φ – семейство предикатов успешности пользовательских запросов. Состояние $\omega \in \Omega$ задается бинарными переменными $\omega_{v,i}$, где $\omega_{v,i} = 1$, если i -я реплика сервиса v работоспособна. В базовой постановке отказы реплик независимы, поэтому P факторизуется по репликам.

$$L_v(\omega) = \mathbb{I}(\sum_{i=1}^{R(v)} \omega_{v,i} \geq 1). \quad (2)$$

Для запроса u предикат $\Phi_u(\omega)=1$ означает, что в подграфе живых вершин существует путь от входной вершины e_u к каждому обязательному элементу целевого множества T_u по синхронным ребрам. Если анализируется немедленный HTTP-ответ, то асинхронные ребра рассматриваются как неблокирующие и исключаются из подграфа связности.

$$A_u = P(\Phi_u(\omega)=1), \quad A_u^{(N)} = (1/N) \sum_{j=1}^N I(\Phi_u(\omega^{(j)})=1). \quad (3)$$

2. Восстановление структуры модели по данным трассировки

Пусть $T=\{t^{(1)}, \dots, t^{(m)}\}$ – множество трасс; в данном восстановлении каждая из них разбирается как дерево спанов трассировки. По множеству T восстанавливаются граф G , отображение τ и вектор репликации R . Для этого отношения «родитель-потомок» между спанами проектируются на уровень сервисов, после чего одинаковые зависимости агрегируются в единый ориентированный граф.

Тип ребра $\tau(e)$ определяется по атрибутам протокола и трассировки: в рассматриваемой модели HTTP- и RPC-взаимодействия трактуются как синхронные, а взаимодействия через обмен сообщениями, включая Kafka, – как асинхронные. Число реплик $R(v)$ оценивается по числу различных идентификаторов экземпляров или контейнеров для сервиса v . Для каждого анализируемого запроса дополнительно задаются входная вершина e_u и целевое множество T_u .

Тем самым решается обратная задача восстановления структуры модели по наблюдаемым исполнениям и строится единое графовое представление системы для последующего расчета доступности.

3. Имитационная оценка доступности и результаты

Точное вычисление доступности на полном пространстве состояний быстро становится вычислительно трудоемким, поэтому используется статистическое имитационное моделирование. В каждом прогоне независимо разыгрываются отказы реплик, строится подграф живых вершин и поиском в ширину или глубину проверяется достижимость обязательных синхронных зависимостей для выбранного запроса.

На DeathStarBench Social Network модель согласуется с натурными экспериментами: коэффициент корреляции Пирсона по всем точкам равен $r \approx 0,992$, а при вероятности отказа реплики 0,3 различие статистически незначимо ($p \approx 0,973$) [4].

На OpenTelemetry Demo, где граф извлекается из данных трассировки OpenTelemetry, модель воспроизводит общую кривую деградации доступности, а при критерии немедленного HTTP-ответа учет асинхронных ребер Kafka меняет оценку не более чем на 0,001 п.п. [5].

Таблица 1 – Сводка результатов валидации модели

Стенд	Метрика	Интерпретация
DeathStarBench Social Network	$r \approx 0,992$; $p \approx 0,973$ при $p_fail=0,3$	Эффект репликации отражается корректно.
OpenTelemetry Demo	$\Delta \leq 0,001$ п.п.	Для немедленного HTTP-ответа влияние ребер Kafka пренебрежимо.

Таким образом, существенная часть наблюдаемой динамики доступности определяется уже структурой зависимостей и репликацией. Учет асинхронных ребер становится принципиальным тогда, когда критерий успеха включает завершение сценария обработки событий, а не только немедленный HTTP-ответ.

Заключение

Предложена единая стохастическая модель связности микросервисной системы для оценки доступности пользовательских запросов при независимых отказах с остановкой. Модель объединяет граф зависимостей, вектор репликации, вероятностную меру на пространстве состояний и предикат успешности, а ее структура восстанавливается по данным распределенной трассировки. Валидация на двух стендах показывает, что модель корректно воспроизводит основную картину деградации доступности и выявляет условия, при которых учет асинхронных ребер существенен. Дальнейшая работа связана с учетом зависимых отказов, повторных попыток, тайм-аутов и моделей надежности пользовательских сценариев.

Список использованных источников

1. Chaos Engineering / A. Basiri, N. Behnam, R. de Rooij [et al.] // IEEE Software. – 2016. – V. 33, №. 3. – P. 35–41.
2. An Open-Source Benchmark Suite for Microservices and Their Hardware-Software Implications for Cloud & Edge Systems / Y. Gan, Y. Zhang, D. Cheng [et al.] // ASPLOS '19. – 2019. – P. 3–18.
3. OpenTelemetry Demo. – URL: <https://opentelemetry.io/docs/demo/> (дата обращения: 28.03.2026).
4. Krasnovsky, A. A. Model Discovery and Graph Simulation: A Lightweight Gateway to Chaos Engineering // ICSE-NIER '26. – 2026. – 5 с.
5. Krasnovsky, A. A. Evaluating Asynchronous Semantics in Trace-Discovered Resilience Models: A Case Study on the OpenTelemetry Demo // arXiv:2512.12314. – URL: <https://doi.org/10.48550/arXiv.2512.12314> (дата обращения: 28.03.2026).

A STOCHASTIC CONNECTIVITY MODEL OF MICROSERVICE SYSTEMS FOR AVAILABILITY ESTIMATION

Anatoly A. Krasnovsky^{1,2}

¹*Innopolis University, Innopolis*

²*MB3R Lab, Innopolis*

e-mail: a.krasnovskii@innopolis.university

Abstract

We present a stochastic connectivity model for endpoint availability in microservice systems under independent fail-stop failures. The model combines a dependency graph, replication map, probability measure on system states, and success predicate. Availability is estimated by Monte Carlo simulation. Validation on DeathStarBench and the OpenTelemetry Demo shows agreement with experiments and clarifies when asynchronous edges matter for immediate HTTP response-based success criteria.

Keywords: microservices, availability, probabilistic graph model, distributed tracing, Monte Carlo.