

DOI: <https://doi.org/10.30898/1684-1719.2023.11.4>

УДК: 004.94

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ И ОТКАЗОУСТОЙЧИВОСТИ СХЕМОТЕХНИЧЕСКИХ САПР НА ОСНОВЕ МЕТОДОВ ДИАКОПТИКИ И АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ МУЛЬТИТЕНАНТНЫМИ КОМПОНЕНТАМИ

В.Н. Гридин¹, В.И. Анисимов^{1,2}, С.А. Васильев¹

¹ Центр информационных технологий в проектировании РАН,
143000, г. Одинцово, ул. Маршала Бирюзова д. 7а

² Санкт-Петербургский государственный электротехнический университет,
197022, г. Санкт-Петербург, ул. Профессора Попова, д. 5

Статья поступила в редакцию 25 августа 2023 г.

Аннотация. Описан подход к построению автоматизированного управления мультитенантных компонентов информационных систем. Отражены и обоснованы основные преимущества использования дискретизирующих мультитенантных оболочек для построения распределенных информационных систем. Приведены методы управления инфраструктурой распределенной системы как кодом. Представлены технологии и их настройки для автоматизации управления на разных стадиях жизненного цикла программной системы, построенной на основе дискретизирующих мультитенантных оболочек. Описано решение проблемы дуального использования менеджера процессов в контейнере на базе операционной системы Linux в ручном и автоматизированном варианте в разрезе управления стадиями жизненного цикла с применением подхода CI/CD. Предложена методика моделирования слабосвязанных электронных схем на основе технологии расчета больших электронных схем по частям, путем выделения слабосвязанных подсхем, между которыми отсутствуют индуктивные связи и выполняется условие концентрации управляющих и управляемых переменных зависимых источников в пределах отдельной подсхемы. Показано, что реализация диакоптического подхода к моделированию больших

слабосвязанных схем существенно увеличивает производительность вычислительного процесса, что особенно актуально при разработке распределенных систем автоматизированного проектирования. Предложено для построения и преобразования описания моделируемой схемы использовать обобщенные сигнальные графы, отображающие уравнения схемы в обобщенной причинно-следственной форме.

Ключевые слова: схемы автоматизированного проектирования, моделирование схем, обобщенный сигнальный граф, методы диакоптики, жизненный цикл, мультитенантность, микросервисная архитектура, CI/CD.

Финансирование: работа выполняется в рамках темы № FFSM-2019-0001.

Автор для переписки: Васильев Сергей Алексеевич info@ditc.ras.ru

Введение

Использование микросервисной архитектуры в распределенных системах, в том числе схемотехнических САПР, позволяет обеспечить массу преимуществ основными из которых являются: масштабируемость, отказоустойчивость, разнообразие технологий. Однако существует и сложности применения такого подхода: сложность управления совокупность сервисов, тестирования развертывания. Обеспечение эффективного использования инфраструктуры достигается использованием виртуализации на уровне мультитенантных контейнеров. Автоматизация управления стадиями жизненного цикла мультитенантных контейнеров, содержащих микросервисы, является актуальной задачей, решение которой обеспечивает высокую степень отказоустойчивости системы и повышения эффективности ее управления.

Наиболее эффективным способом повышения производительности программного обеспечения схемотехнических САПР для моделирования больших схем является разделение моделируемой схемы на составляющие подсхемы с последующим формированием и преобразованием для каждой из подсхем необходимого математического описания. Общее решение для всей схемы получается путем соединения полученных частных решений

для подсхем [1-3]. При этом не требуется выполнять формирование полной системы уравнений для всей схемы, достаточно сформировать и соответствующим образом обработать уравнения для ее элементарных подсхем, которые могут быть сделаны настолько малыми, насколько это требуется для расчета конкретной схемы. Использование диакоптического подхода для расчета больших слабосвязанных электронных схем является особенно актуальным при разработке распределенных систем автоматизированного проектирования, поскольку при работе с сети Интернет необходимо обеспечить достаточно высокое быстродействие распределенной системы чтобы минимизировать время взаимодействия пользователя с сетью [4-7].

Автоматизация управления мультитенантными компонентами информационной системы

Микросервисная архитектура представляет собой подход к разработке программного обеспечения, при котором приложение разбивается на небольшие, слабо связанные и независимо развертываемые компоненты, называемые микросервисами. Каждый микросервис выполняет определенные функции и может разрабатываться, тестироваться и развертываться отдельно.

Для обеспечения мультитенантности микросервисов используются контейнеры, обладающие рядом преимуществ, для развертывания и управления микросервисами [8]:

- **Изолированность:** Каждый микросервис может быть упакован в свой собственный контейнер, что обеспечивает изоляцию от других микросервисов и окружения. Это позволяет избежать конфликтов между зависимостями и обеспечивает надежную работу каждого сервиса в отдельности.
- **Переносимость:** Контейнеры предоставляют высокую степень переносимости, поскольку все зависимости и конфигурации приложения упакованы вместе с приложением. Это облегчает развертывание микросервисов на различных платформах и облаках без необходимости вносить изменения в код приложения.

- **Масштабируемость:** Контейнеры обеспечивают горизонтальное масштабирование, что позволяет запускать несколько экземпляров одного и того же контейнера, чтобы справиться с повышенной нагрузкой. Это позволяет эффективно управлять ресурсами и обеспечить высокую производительность.
- **Удобство управления:** Контейнеры предоставляют простой и единый способ управления приложениями и их зависимостями. Оркестраторы контейнеров, такие как Kubernetes, облегчают автоматизацию развертывания, масштабирования и управления микросервисами.
- **Более эффективное использование ресурсов:** Контейнеры используют общую операционную систему и разделяют ядро хоста, что позволяет более эффективно использовать ресурсы сервера по сравнению с виртуальными машинами.
- **Обновления без простоя:** Контейнеры позволяют проводить обновления приложения без прерывания его работы. При обновлении контейнера новая версия приложения запускается, а старая останавливается, что позволяет избежать простоя сервиса.
- **Инфраструктура как код:** Контейнеры позволяют определить инфраструктуру как код, что облегчает автоматизацию развертывания и управления инфраструктурой.
- **Изоляция сети:** Контейнеры могут иметь свои собственные сетевые пространства, что обеспечивает безопасное взаимодействие между микросервисами и избегает конфликтов сетевых портов.

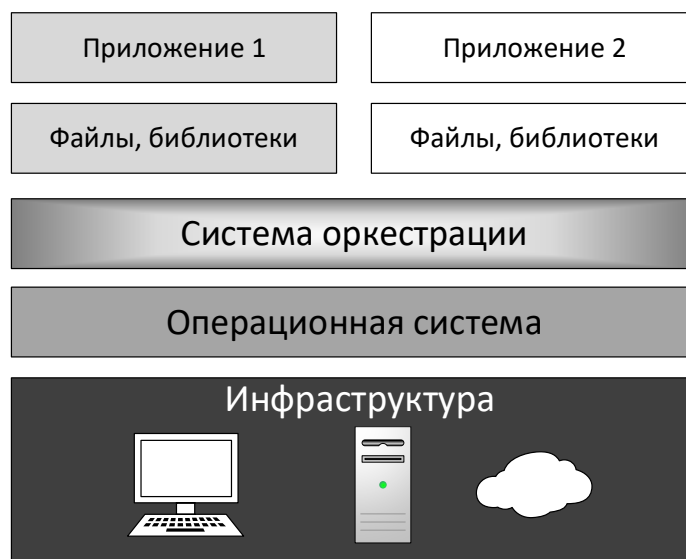


Рис. 1. Схема микросервисной архитектуры на основе контейнеров.

Для обеспечения автоматизации управления микросервисами в рамках дискретизирующих мультитенантных оболочек (рис. 1.) применяется механизм комбинации непрерывной интеграции и непрерывного развертывания программного обеспечения в процессе разработки CI/CD (Continuous Integration/Continuous Deployment). Каждый из микросервисов входящих в систему имеет свой отдельный жизненный цикл и жизненный цикл в составе программного продукта. Механизм CI/CD является важной частью жизненного цикла микросервиса и системы, играя ключевую роль на различных этапах этого цикла. Жизненный цикл приложения включает в себя этапы разработки, тестирования, развертывания, эксплуатации и поддержки приложения от начала до конца его существования [9-10]:

- **Разработка и кодирование.** CI обеспечивает автоматическую интеграцию кода в общий репозиторий после каждого коммита. Это позволяет команде получать обратную связь о работоспособности кода и быстро обнаруживать конфликты или ошибки.
- **Сборка и тестирование.** После каждого коммита CI-сервер автоматически собирает приложение и запускает автоматизированные тесты, чтобы проверить его работоспособность. CD-пайплайн включает тестирование на различных окружениях (например, тестовом, стейджинге и продакшене) для обеспечения согласованности поведения приложения.

- Развертывание и интеграция. CD обеспечивает непрерывное развертывание приложения на продакшене или других окружениях с минимальным вмешательством разработчиков.
- Эксплуатация и мониторинг. CD может включать мониторинг приложения в реальном времени, чтобы автоматически реагировать на события и обеспечивать надежность работы.
- Обратная связь и обновления. CI/CD предоставляет обратную связь о работоспособности и производительности приложения, что помогает разработчикам быстро выявлять и исправлять ошибки и выпускать обновления.
- Поддержка и обслуживание. CI/CD позволяет автоматизировать задачи поддержки, такие как исправление багов, обновление зависимостей и перенос приложения на новые версии платформы.

Использование CI/CD в жизненном цикле приложения позволяет автоматизировать и оптимизировать процессы разработки и развертывания, сокращает время отклика на изменения и повышает общую надежность и качество приложения. Это становится особенно важным в современных быстро развивающихся проектах и в условиях микросервисной архитектуры, где многочисленные компоненты требуют постоянных обновлений и координации.

Современную экосистему программного обеспечения, в разрезе его жизненного цикла, можно представить в виде микросервисных контейнеров на базе операционной системы Linux, платформы node.js, менеджер процессов PM2 и системы контроля версий на основе Git, например, GitLab со встроенным механизмом CI/CD (или отдельное программное обеспечение для обеспечения механизма CI/CD Jenkins). Наиболее оптимальным каналом взаимодействия инструментов CI/CD с контейнерами Linux является SSH соединение. С целью обеспечения безопасности и контроля действия в операционной системе Linux создается специальный пользователь с особыми правами и ключами авторизации для процессов CI/CD и администраторов микросервисов. В таком случае появляется проблема управления конкретным процессом

микросервиса запущенного в контейнере от лица различных пользователей (CI/CD инструмента и администратора, а также других пользователей по необходимости). По умолчанию PM2 устанавливается и разворачивает конфигурацию в личной папке пользователя, поэтому каждый пользователь управляет своими процессами что может привести к неожиданному поведению и проблемам при управлении приложением.

Чтобы предоставить общий доступ к файлам конфигурации и приложениям, для взаимодействия с одним и тем же процессом PM2 от нескольких пользователей, следует выполнить следующие шаги:

- Создать общую директорию, которая будет служить общим хранилищем для всех файлов, необходимых для запуска приложения. Например, `/shared_pm2`.
- Предоставьте права доступа для этой директории таким образом, чтобы она была доступна для чтения и записи для всех пользователей, которые будут управлять приложением. Например, можно использовать команду `chmod` на Linux или macOS: `chmod -R 777 /path/to/shared_pm2`.
- Разместите файлы конфигурации PM2 в эту общую директорию `/shared_pm2`.
- Запустите процесс PM2. Каждый настроенный соответствующим образом пользователь теперь может использовать общую директорию для запуска процесса PM2 и указать путь к файлу конфигурации в этой директории. Например, `pm2 start /path/to/shared_pm2/...`

Выполнив соответствующие настройки, настроенные пользователя, смогут управлять процессами PM2 (например, перезапускать, останавливать и т.д.) с помощью команд PM2, указывая путь к файлу конфигурации. Обратите внимание, что предоставление общего доступа к файлам в общем случае может повлечь за собой проблемы безопасности, убедитесь, что вы принимаете все необходимые меры для обеспечения безопасности и ограничения доступа к общей директории только необходимым пользователям.

Приведения методов управления для случая с общей директорией файлов конфигурации и стандартной пользовательской папкой можно использовать команду alias для создания альтернативных версий стандартных команд alias pm2='env HOME=/.../.../... pm2'.

Математическое описание больших электронных схем на основе диакоптики

В моделируемой схеме всегда можно провести декомпозицию ее структуры путем выделения слабосвязанных подсхем, между которыми отсутствуют индуктивные связи и выполняется условие концентрации управляющих и управляемых переменных зависимых источников в пределах отдельной подсхемы (рис. 2).

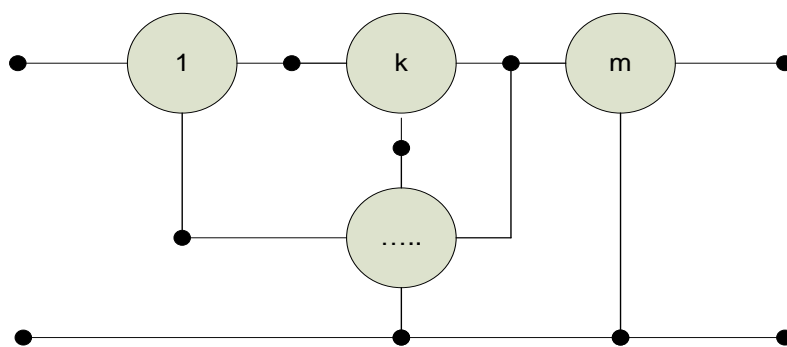


Рис. 2. Декомпозиция слабосвязанной схемы.

Для каждой выделенной k -й подсхемы можно ввести внутренние переменные x_i и образовать для каждой подсхемы вектор внутренних переменных $X_k = [..., x_i, ...]^t$. Для каждого узла схемы можно ввести узловую переменную v_p , определяющую состояние этого узла, и из всех введенных узловых переменных образовать вектор узловых потенциалов $X_0 = V_0 = [..., v_p, ...]^t$. Если провести нумерацию переменных таким образом, что в начале нумеруются переменные отдельных подсхем, а на заключительном этапе узловые переменные связи между подсхемами, то уравнения моделируемой схемы будут содержать матрицу с окаймленной блочно-диагональной структурой.

W_{11}					W_{10}

		W_{kk}			W_{k0}
		
				W_{mm}	W_{m0}
W_{0l}	...	W_{0k}	...	W_{0m}	W_{00}

X_1
...
X_k
...
X_m
X_0

S_1
...
S_k
...
S_m
S_0

$$= 0 \quad (1)$$

Система уравнений для k-й подсхемы имеет вид:

$$W_{kk} X_k + W_{k0} X_0 + S_k = 0, \quad k = \overline{1, m}. \quad (2)$$

Отсюда можно получить выражения для вектора внутренних переменных отдельных подсхем:

$$X_k = -W_{kk}^{-1} W_{k0} X_0 - W_{kk}^{-1} S_k, \quad k = \overline{1, m}.$$

Подставляя это выражение в последнюю блочную строку окаймленной матрицы в уравнении (1), получим уравнение для определения вектора переменных узлов связи:

$$\overline{W}_{00} X_0 + \overline{S} = 0,$$

где $\overline{W}_{00} = W_{00} - \sum_{k=1}^m W_{0k} \overline{W}_{k0};$

$$\overline{S}_0 = S_0 - \sum_{l=1}^m W_{0l} \overline{S}_l;$$

$$\overline{W}_{k0} = W_{kk}^{-1} W_{k0};$$

$$\overline{S}_k = W_{kk}^{-1} S_k.$$

Таким образом, вместо решения общего уравнения схемы $WX + S = 0$, что требует вычисления обратной матрицы для общей матрицы схемы W , необходимо выполнить обращение матриц W_{kk} и W_{00} , порядок которых может быть значительно снижен путем выбора при декомпозиции исходной схемы достаточно большого числа подсхем. При этом нет необходимости в составлении полной системы уравнений, требуется только сформировать уравнения для ее подсхем и для переменных узлов связи.

Однако отмеченные преимущества диакоптического подхода к расчету больших слабосвязанных схем будут сохраняться только при условии

неизменности сохранения описания моделируемой схемы в виде блочно-диагональной окаймленной структуры (1). Вместе с тем, в реальных условиях в процессе проектирования схемы неизбежно будут появляться новые переменные, что неизбежно приведет к нарушению первоначальной блочно-диагональной структуры и потребует полной перекодировки переменных, чтобы сохранить окаймленную структуру описания схемы.

Для решения указанной проблемы целесообразно использовать топологическое описание моделируемой схемы при помощи обобщенных сигнальных графов [11], которые отображают систему исходных уравнений в обобщенной причинно-следственной форме и содержат взвешенные вершины. Помимо наглядного представления о структуре связей между подсхемами описание задачи моделирования на основе обобщенных сигнальных графов позволяет легко решить проблему выбора последовательности нумерации внутренних переменных подсхем и переменных узлов связи моделируемой схемы.

В обобщенной причинно-следственной форме уравнение для k -й подсхемы может быть записано в виде:

$$W_{kk} X_k = -W_{k0} X_0 - C_k S_k = 0, \quad k = \overline{1, m}, \quad (3)$$

где W_{kk} , W_{k0} , – матрицы параметров k -подсхемы, S_k – вектор задающих источников k -подсхемы, C_k – матрица соединений задающих источников k - подсхемы.

Аналогично, для узлов связи между подсхемами уравнение в обобщенной причинно-следственной форме имеет вид

$$W_{00} X_0 = -\sum_{k=1}^m W_{0k} X_k - C_0 S_0, \quad (4)$$

где W_{00} , W_{0k} – матрицы параметров узлов связи между подсхемами, S_0 – вектор задающих источников, C_0 – матрица соединений задающих источников, подключенных к узлам связи.

Уравнениям (3) и (4) соответствует приведенный на рис. 3 обобщенный сигнальный граф.

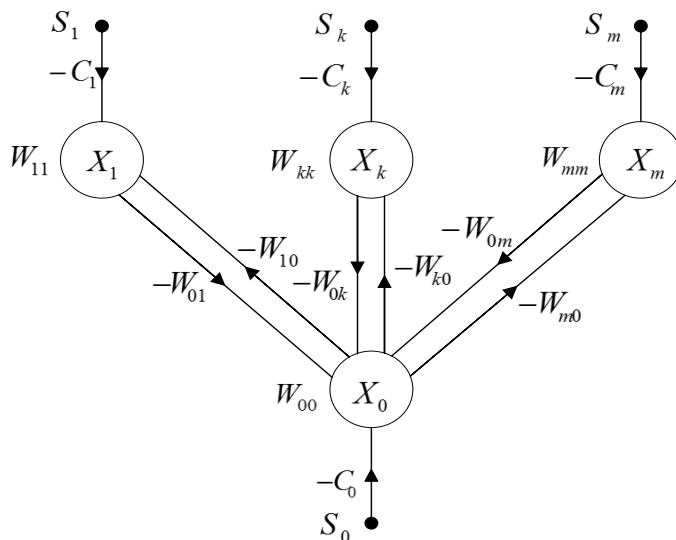


Рис. 3. Обобщенный сигнальный граф моделируемой схемы.

Построенный обобщенный сигнальный граф отображает взаимодействие между отдельными подсхемами и переменными связи моделируемой схемы и позволяет обеспечить независимую нумерацию переменных и обработку описания отдельных подсхем, а также подключение новых подсхем к узлам связи.

Умножая уравнение (2) на обратную матрицу W_{kk}^{-1} , можно привести это уравнение к виду

$$X_k = -\bar{W}_{k0} X_0 - \bar{C}_k S_k, \quad k = \overline{1, m}, \quad (5)$$

где $\bar{W}_{k0} = W_{kk}^{-1} W_{k0}$;

$$\bar{C}_k = W_{kk}^{-1} C_k.$$

Для построения заключительного топологического описания задачи подставим уравнение (5) в уравнение (4), что дает:

$$\bar{W}_{00} X_0 = \bar{S}_0, \quad (6)$$

где $\bar{W}_{00} = W_{00} - \sum_{k=1}^m W_{0k} \bar{W}_{k0}$;

$$\bar{S}_0 = -C_0 S_0 + \sum_{k=1}^m W_{0k} \bar{C}_k S_k.$$

Уравнениям (5) и (6) соответствует обобщенный сигнальный граф, приведенный на рисунке 4.

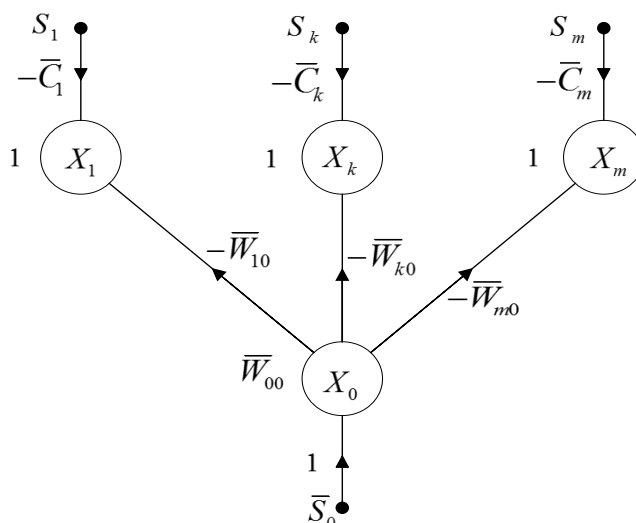


Рис. 4. Заключительное представление топологического описания схемы.

По виду этого слабосвязанного графа можно легко найти значения переменных вектора связей X_0 и векторов X_k внутренних переменных подсхем.

Практическая реализация построения заключительного топологического описания возможна непосредственно по виду исходного обобщенного сигнального графа моделируемой схемы путем топологического исключения из него всех ветвей, исходящих из подграфов, содержащих подмножество вершин X_k подсхем, и нормирования значений весовых функций обобщенных взвешенных вершин $W_{kk} = 1$. При выполнении указанных операций будут пересчитываться весовые функции всех ветвей, заходящих в обобщенные вершины X_k .

Следует иметь в виду, что построение заключительной структуры топологического описания схемы возможно только при выполнении условия существования для всех подсхем обратной матрицы W_{kk}^{-1} . Очевидно, что для этого необходимо выполнение условия неравенства нулю определителя Δ матриц W_{kk} для всех подсхем. Эти условия обычно выполняются, если при декомпозиции исходной схемы в качестве отдельных подсхем выделять законченные функциональные блоки, каждый из которых выполняет вполне

определенную задачу. В исключительных случаях может понадобиться выполнение отдельного предварительного анализа устойчивости функционирования отдельных подсхем [12,13].

Топологическое представление описания моделируемой схемы в виде обобщенного сигнального графа предоставляет широкие возможности для анализа топологической структуры графа и всех его элементарных графов g_i . Каждый элементарный граф содержит совокупность некасающихся контуров и взвешенных вершин (включая взвешенные вершины с нулевым весом). Определитель обобщенного сигнального графа Δ можно вычислить по топологической формуле $\Delta = \sum \delta_i$, где δ_i – определитель элементарного графа, равный произведению весов входящих в него некасающихся контуров и взвешенных вершин [11].

Эффективность моделирования больших схем на основе диакоптического подхода можно оценить коэффициентом экономии оперативной памяти и степенью повышения быстродействия вычислительных операций.

Пусть V_D – объем памяти, которая требуется для расчета схемы с использованием методов диакоптики, V – объем памяти, требуемый для расчета этой схемы обычными методами без выполнения декомпозиции моделируемой структуры. Если схема разделена на m подсхем, число переменных в каждой подсхеме равно n_k , а число переменных связи составляет n_0 , то коэффициент экономии оперативной памяти $\alpha = V/V_D$ при диакоптическом подходе можно определить выражением

$$\alpha = \frac{(n_0 + \sum_{k=1}^m n_k)^2}{\max_k n_k^2 + 2n_0 \sum_{k=1}^m n_k + n_0^2}$$

Если схема разделена на 10 одинаковых подсхем ($m = 10$) и число переменных в каждой подсхеме равно числу переменных связи $n_k = n_0$, то получим значение коэффициента экономии оперативной памяти $\alpha = 6$.

Для оценки степени повышения быстродействия на основе диакоптического подхода следует иметь в виду, что время решения уравнений определяется числом мультипликативных операций и пропорционально величине $n^3/3$ для метода Гаусса и $n^3/2$ для метода Гаусса-Жордана. Следовательно, эффективность по быстродействию может быть определена выражением

$$\beta = \frac{(n_0 + \sum_{k=1}^m n_k)^3}{\sum_{k=1}^m n_k^3 + b n_0^3 \sum_{k=1}^m n_k + n_0^3},$$

где $b = 1/3$ или $b = 1/2$ в зависимости от выбранного метода решения уравнений.

Если моделируемая схема разбита на 10 одинаковых подсхем и число переменных в каждой подсхеме равно числу переменных связи, т.е. $n_k = n_0$, то значение степени повышения быстродействия при решении задачи методами диакоптики $\beta = 30$.

Таким образом, диакоптический подход к решению задачи моделирования больших слабосвязанных схем обеспечивает существенное увеличение коэффициента экономии оперативной памяти и значительно повышает быстродействие при решении задач моделирования.

Заключение

Описанный подход к построению микросервисной архитектуры позволяет разбить информационную систему, в частности систему автоматизации схемотехнического проектирования, на независимые компоненты, обеспечивая изоляцию, переносимость, масштабируемость и эффективное использование ресурсов. Контейнеры также обеспечивают удобное управление и обновление инфраструктуры, используя парадигму инфраструктура – код. Для автоматизации управления микросервисами используется подход CI/CD, включающий этапы разработки, тестирования, развертывания, эксплуатации и обновлений, и обеспечивающий эффективный и гибкий способ создания и управления компонентами информационной системы.

Показано, что диакоптический подход к моделированию больших слабосвязанных схем на основе декомпозиции исходной схемы на ряд составляющих подсхем приводит к существенному повышению производительности системы автоматизации схемотехнического проектирования. Использование методов диакоптики обеспечивает значительную экономию затрат оперативной памяти и существенно снижает время решения задач моделирования больших слабосвязанных схем.

Литература

1. Г. Крон. Исследование сложных схем по частям – диакоптика, пер с англ. Издательство «Наука», Главная редакция физико-математической литературы. М.: 1972.
2. Х. Хэпп. Диакоптика и электрические цепи, пер. с англ. -М.: Изд. «Мир» 1974. – 342 с.
3. Б.В. Баталов, Ю.Б. Егоров, С.Г. Русаков. Основы математического моделирования больших интегральных схем на ЭВМ. -М.: Радио и связь, 1982, – 168 с.
4. В.И. Анисимов, В.Н. Гридин. Методы построения схем автоматизированного проектирования на основе Интернет-технологий и компактной обработки разреженных матриц. //Информационные технологии в проектировании и производстве. 2009. №1. с. 3-7.
5. О.С. Коваленко, В.М. Курейчик. Обзор проблем и состояний облачных вычислений и сервисов. Известия ЮФУ. Технические науки. № 7, 2012. с.146-153
6. В.Н. Гридин, Г.Д. Дмитриевич, Д.А. Анисимов. Построение схем автоматизированного проектирования на основе Web-технологий // Информационные технологии. 2011. №5. с. 23-27.
7. Д.А. Анисимов. Методы построения схем автоматизации схемотехнического проектирования на основе веб-сервисов // Известия СПбГЭТУ «ЛЭТИ» 2012. №10. с. 56-61

8. N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina, “Microservices: Yesterday, today, and tomorrow,” in Present and Ulterior Software Engineering, M. Mazzara and B. Meyer, Eds. Switzerland: Springer International Publishing AG, 2017, pp. 195–216.
9. C. Esposito, A. Castiglione, and K.-K. R. Choo, “Challenges in delivering software in the cloud as microservices,” IEEE Cloud Computing, vol. 3, no. 5, pp. 10–14, Sep. 2016
10. C. E. da Silva, Y. de Lima Justino, and E. Adachi, “SPReaD: Serviceoriented process for reengineering and DevOps,” Service Oriented Computing and Applications, 2021.
11. В.И. Анисимов. Топологический расчет электронных схем. Л.: «Энергия», 1977, 238 с.
12. В.Н. Гридин, В.Б. Михайлов, Л.Б. Шустерман. Численно-аналитическое моделирование радиоэлектронных схем. //Наука, 2008, 339 с.
13. Б.А. Калабеков, И.Ю. Лapidус, В.М. Малафеев. Методы автоматизированного расчета электронных схем в технике связи. – М.: «Радио и связь», 1990. 272 с.

Для цитирования:

Гридин В.Н., Анисимов В.И., Васильев С.А. Повышение производительности и отказоустойчивости схемотехнических САПР на основе методов диакоптики и автоматизации управления мультитенатными компонентами. // Журнал радиоэлектроники. – 2023. – №. 11. <https://doi.org/10.30898/1684-1719.2023.11.4>