

Автоматизация DevOps в распределенной вычислительной среде применительно к научным приложениям

Евгений Николаев

Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134

Аннотация

Поддержка DevOps применительно к научным приложениям в гетерогенной распределенной вычислительной среде (ГРВС) является сложной и актуальной проблемой. Известен широкий спектр инструментов, реализующих отдельные этапы DevOps. К сожалению, готовое решение по их согласованному автоматизированному использованию применительно к научным приложениям, рабочие процессы которых выполняются в ГРВС с помощью систем управления прохождением заданиями, автору не известно. В этом контексте предложен подход к автоматизации процессов непрерывной интеграции и контейнеризации прикладного и системного программного обеспечения, используемого при выполнении рабочих процессов сервис-ориентированных научных приложений на вычислительном кластере. В рамках подхода применяется связка систем Ansible, Docker и Kubernetes. Эксперименты по апробации подхода показали его работоспособность и эффективность, а также существенное сокращение времени развертывания и настройки необходимого прикладного и системного программного обеспечения.

Ключевые слова

Сервис-ориентированные приложения, рабочие процессы, гетерогенная распределенная вычислительная среда, DevOps

1. Введение

Научная проблема поддержки DevOps применительно к научным приложениям в гетерогенной распределенной вычислительной среде (ГРВС) заключается в наличии существенных накладных расходов на доставку, развертывание, настройку, тестирование и запуск программного обеспечения (ПО) в ручном режиме [1]. Более того эти расходы стремительно возрастают при увеличении числа узлов среды. Ключевыми аспектами, влияющими на степень сложности решения данной проблемы, являются следующие:

- различия в характеристиках (архитектуре, производительности, размере оперативной и дисковой памяти, интерконнекте и др.) и программном окружении (ОС, версиях библиотек, зависимостях и т. п.) узлов среды;
- снижение надежности и отказоустойчивости программно-аппаратных средств среды, на которых ПО развернуто в ручном режиме, и в следствии этого возрастание риска возникновения нештатных аварийных ситуаций при выполнении этого ПО;
- специфика решения задач для каждой предметной области с помощью разворачиваемого ПО.

Разработка автоматизированной технологии управления жизненным циклом ПО в ГРВС позволит достичь следующих эффектов: существенно снизить накладные расходы, связанные с

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: zhenya_nikolaev_1995@list.ru (Евгений Николаев)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2025 Workshop Proceedings

DOI: 10.47350/ICCS-DE.2025.32

ручной доставкой, развертыванием, настройкой, тестированием и запуском ПО; обеспечить воспроизводимость экспериментов в ГРВС; повысить надежность и отказоустойчивость программно-аппаратных средств узлов среды; адаптировать процессы DevOps под специфику решения задач в разных предметных областях.

В рамках данного исследования осуществлены выбор и интеграция DevOps-инструментов, определена технология непрерывной интеграции и контейнеризации ПО, используемого при выполнении рабочих процессов сервис-ориентированных научных приложений на вычислительном кластере, и автоматизирована настройка ресурсов вычислительного кластера.

2. Методы и средства

Разработка научных приложений и их рабочих процессов осуществляется с помощью инструментального комплекса FDE-SWFs [2]. Для реализации отдельных этапов DevOps применительно к системному и прикладному ПО, необходимому для выполнения рабочих процессов, может использоваться ряд известных инструментов [3]. При этом их комплексное применение, как правило, определяется в каждом случае индивидуально, исходя из спектра поддерживаемых этапов DevOps, используемых вычислительных ресурсов, применяемого ПО и др. В рамках данного исследования выбрана связка систем «Ansible [4] + Docker [5] + Kubernetes [6]», использование которой рекомендовано в [1]. Настройка узлов, размещение на них ПО и модификация конфигурационных файлов осуществляется с помощью Ansible. Система Docker применяется для создания контейнеров, запуск которых и управление ими выполняется системой Kubernetes. В общем случае подготовка и проведение экспериментов в рамках предложенного подхода включает следующие основные этапы (рис. 1):

1. Подготовка узлов ГРВС и сбор информации о них с помощью системы Ansible, включая установку Kubernetes.
2. Интеграция системного ПО из Git-репозитория в базовый образ (БО) системы Docker. БО создается под ОС Linux. Он содержит компоненты диспетчера и СУПЗ.
3. Построение набора специализированных образов $CO_1, \dots, CO_n$ путем адаптации БО к особенностям используемых ресурсов среды путем настройки конфигурационных файлов с помощью системы Ansible. Отдельный образ содержит управляющий компонент центрального узла среды – Central Manager (CM). CM управляет рабочими узлами среды – Execute.
4. Интеграция прикладного ПО (модулей приложения, необходимых внешних библиотек из Git-репозитория и компонентов системы мониторинга вычислительных ресурсов и процессов), подключение временной базы данных (ВБД) для хранения промежуточных расчетов в $CO_1, \dots, CO_n$ и сборка на их основе интегрированных образов (ИО) $IO_1, \dots, IO_n$.
5. Размещение $IO_1, \dots, IO_n$ на ресурсах с помощью системы Ansible.
6. Проверка соединений и тестирование ресурсов.
7. Запуск контейнеров на основе $IO_1, \dots, IO_n$, их конфигурирование и оркестрация на ресурсах с помощью системы Kubernetes.
8. Запуск и выполнение рабочего процесса диспетчером рабочих процессов FDE-SWFs в ГРВС и его мониторинг.
9. Сохранение результатов расчетов и освобождение ресурсов.

3. Экспериментальный анализ

В качестве примера применения предложенного подхода к автоматизации DevOps рассматривается подготовка и проведение экспериментов по решению задачи глобального анализа чувствительности [7]. Приложение для решения задачи включает один научный рабочий процесс (рис. 2), состоящий из девяти параметров и реализуемый с помощью пяти модулей, представленных в виде сервисов. Вычислительная среда включает три узла со следующими характеристиками: операционная система – Ubuntu 22.04, размер оперативной памяти – 4012 Мб, число процессоров – 5, объем видеопамати – 16 Мб. Управление заданиями осуществляется с помощью СУПЗ HTCondor [8]. Сопоставление оценок накладных расходов с накопительным

итогом на подготовку и проведение экспериментов, приведенных на рис. 2, показывает их сокращение за счет автоматизации DevOps в сравнении с ручным режимом более чем в два с половиной раза. Наибольшее сокращение достигается на этапах настройки конфигурационных файлов HTCondor и обеспечения сетевого взаимодействия между узлами кластера.

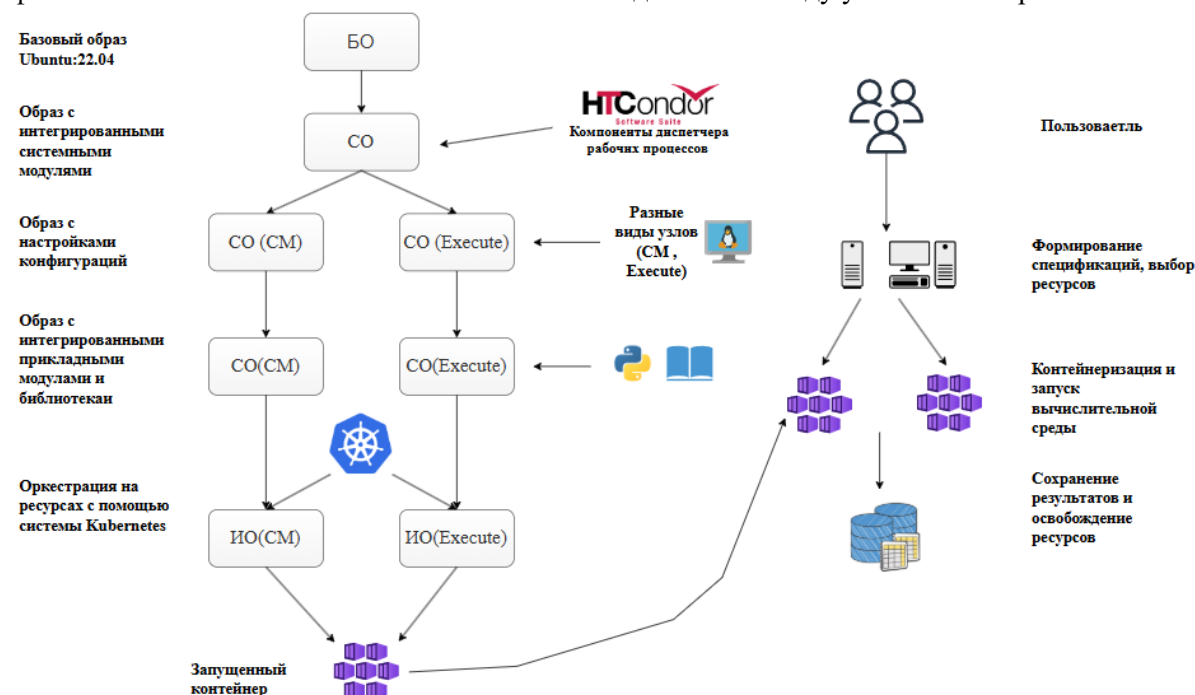


Рисунок 1: Основные этапы подготовки и проведения экспериментов

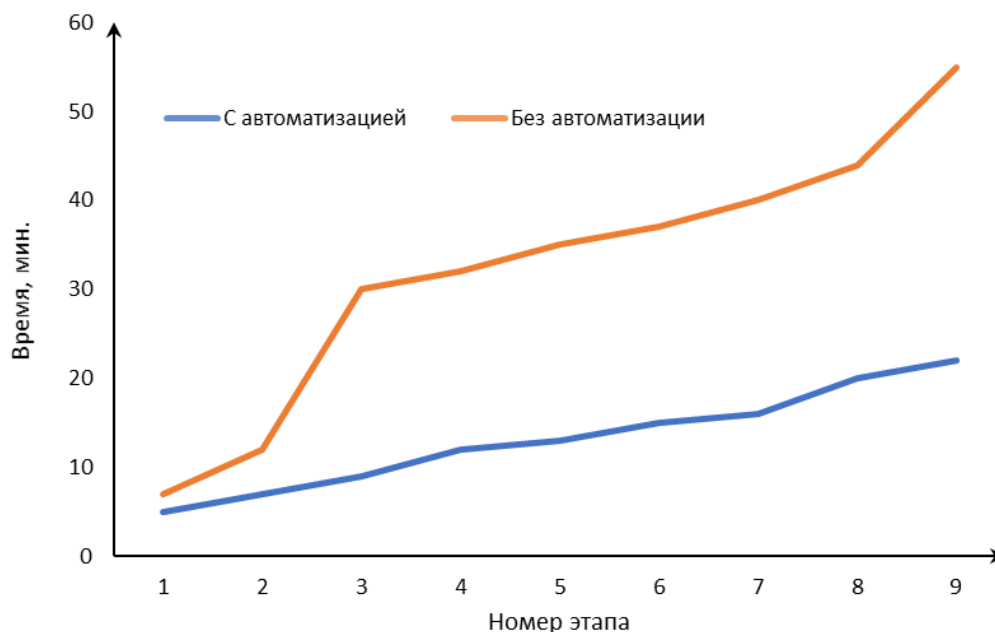


Рисунок 2: Оценка накладных расходов с накопительным итогом на подготовку и проведение экспериментов

4. Заключение

Предложен подход к автоматизации непрерывной интеграции и контейнеризации ПО, необходимого для выполнения рабочих процессов сервис-ориентированных приложений, с помощью связки «Ansible + Docker + Kubernetes». Работоспособность и эффективность применения данной связки продемонстрированы на реальном примере решения практической

задачи. Показано, что автоматизация непрерывной интеграции и контейнеризации ПО на основе предложенного подхода обеспечивает существенное сокращение времени подготовки и проведения экспериментов.

5. Благодарности

Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации, проект № FWEW-2021-0005 «Технологии разработки и анализа предметно-ориентированных интеллектуальных систем группового управления в недетерминированных распределенных средах» (рег. № 121032400051-9) и грант № 075-15-2024-533 на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (рег. № 124052100088-3).

6. Список использованных источников

- [1] Р. Костромин. Подход к развертыванию программного обеспечения вычислительного кластера по облачной модели, Материалы VI Международного семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2024, 1-5 июля 2024 г., г. Иркутск, Россия), Иркутск: Изд-во ИДСТУ СО РАН, 2024, с. 135–140. doi:0.47350/ICCS-DE.2024.18.
- [2] М. Л. Воскобойников, А. Г. Феоктистов, А. Н. Черных. Разработка и применение сервис-ориентированных научных приложений в инструментальном комплексе FDE-SWFs, Труды ИСП РАН, 2024 36(6) 195–214. doi:10.15514/ISPRAS-2024-36(6)-11.
- [3] Инструменты DevOps. URL: <https://learn.microsoft.com/en-us/devops>.
- [4] Ansible. URL: <https://www.redhat.com/en/ansible-collaborative>.
- [5] Docker. URL: <https://www.docker.com/>.
- [6] Kubernetes. URL: <https://kubernetes.io/>.
- [7] A. Feoktistov, A. Edelev, A. Tchernykh, S. Gorsky, O. Basharina, E. Fereferov. An Approach to Implementing High-Performance Computing for Problem Solving in Workflow-based Energy Infrastructure Resilience Studies, *Computation*, 2023, vol. 11, no. 12, pp. 243. doi:10.3390/computation11120243.
- [8] HTCondor. URL: <https://htcondor.com/index.html>.