



# Evolving the system for managing a computational cluster through containerization

*Развитие системы управления вычислительным кластером посредством контейнеризации*

*Роман Олегович Костромин, Александр Геннадьевич Феоктистов*

# Вычислительный кластер «Академик В.М. Матросов»

Ввод в эксплуатацию: 2012 г.

Модернизация: 2017, 2023 гг.

Разработчики/поставщики оборудования: «Т-Платформы», «Ниагара Компьютерс», «Карма Групп» и др.

**242** процессора x86\_64 (CPU): **4112** CPU-ядер

**4** графических процессора (GPU): **1728** тензорных ядра, **27648** CUDA-ядер

Суммарная пиковая  
производительность: **171,21 TFlops**

Реальная производительность (HPL): **136,56 TFlops**



# Цикл работы с кластером

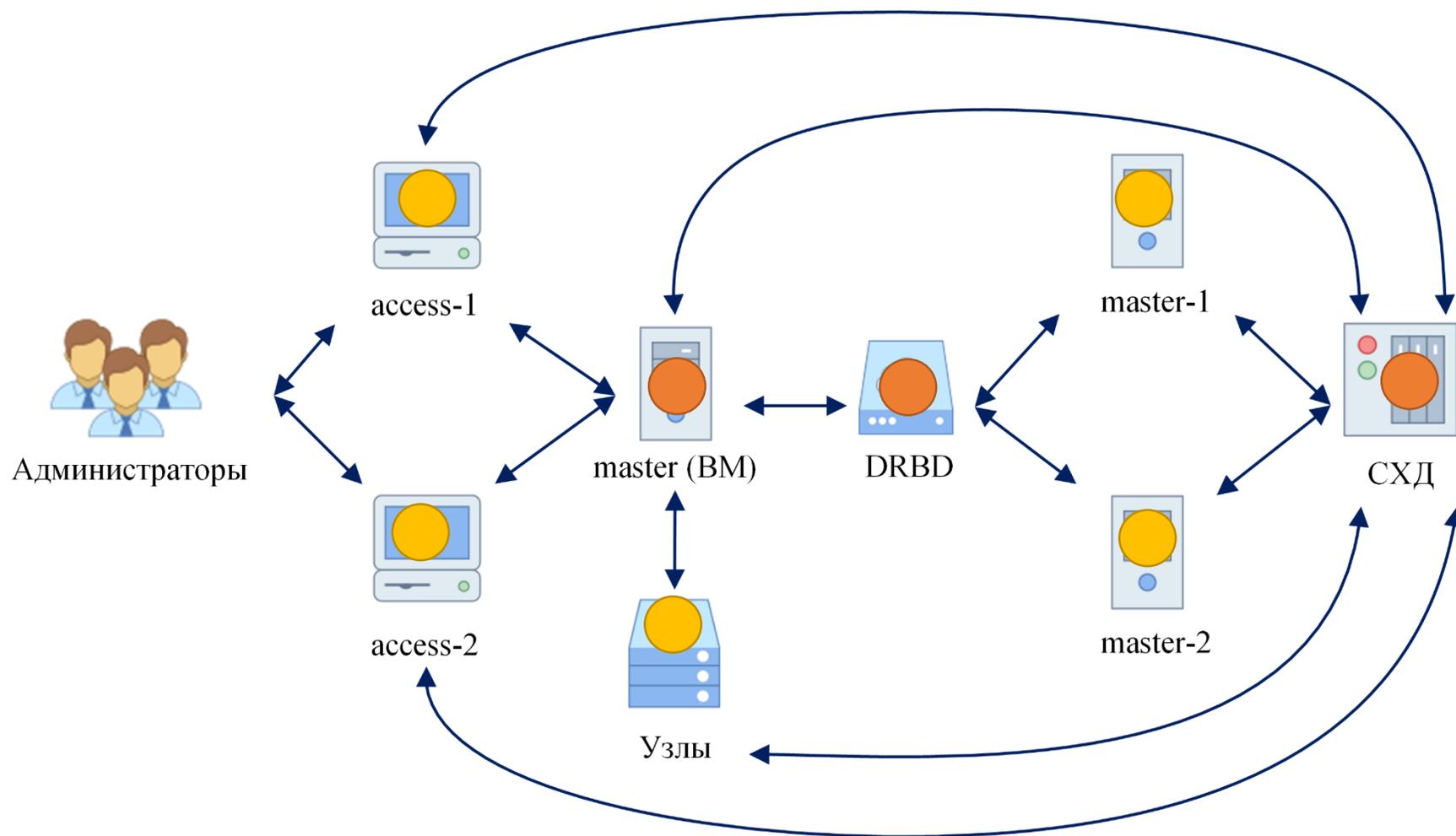
Цикл работы пользователя с кластером состоит из следующих этапов:

- Удаленный вход на кластер.
- Копирование данных между кластером и компьютером пользователя.
- Редактирование исходных текстов программ.
- Компиляция программ.
- Запуск задач и работа с очередью.
- Завершение сеанса.



Общая схема работы с кластером.

# Кластер с точки зрения администратора и точки отказа



Точки отказа кластера

# Гетерогенная вычислительная среда

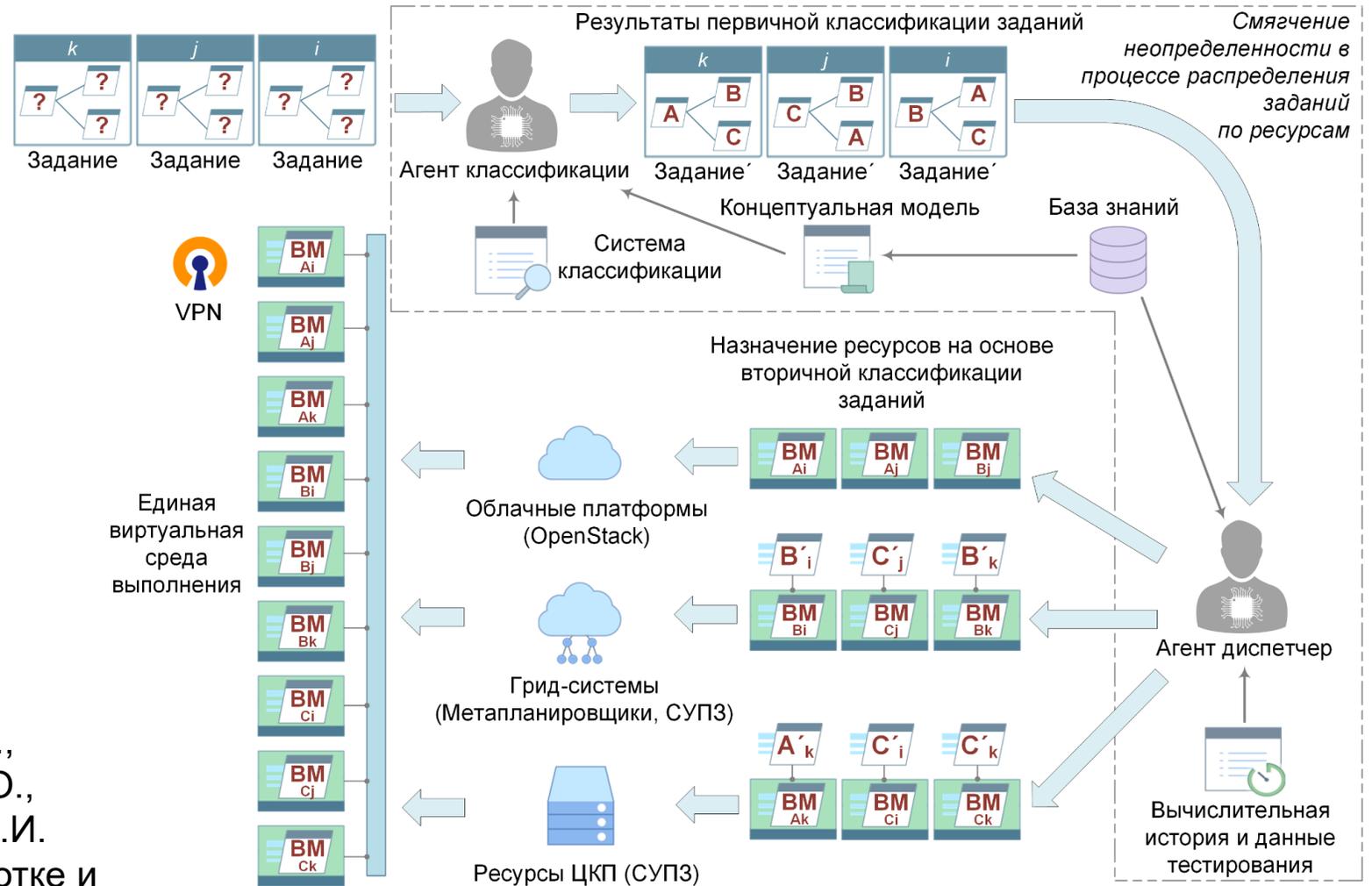


Схема интеграции разнородных ресурсов

Черных А.Н., Бычков И.В., Феоктистов А.Г., Горский С.А., Сидоров И.А., Костромин Р.О., Еделев А.В., Зоркальцев В.И., Аветисян А.И. Смягчение неопределенности при разработке и применении научных приложений в интегрированной среде // Труды ИСП РАН. 2021. Т. 33. № 1. С. 151-172.

# Рассуждения о повышении надежности кластера

## Предложение 2:

- временный перенос master на отдельный узел (бэкап всего!);
- очистка master1 и master2, обновление дисков в аппаратном RAID, установка среды контейнеризации (Kubernetes, Docker Swarm);
- выделение ролей master в независимые сущности (контейнеры);
- выделение ролей access в независимые сущности (контейнеры);
- реплицирование контейнеров по кластеру контейнеризации;

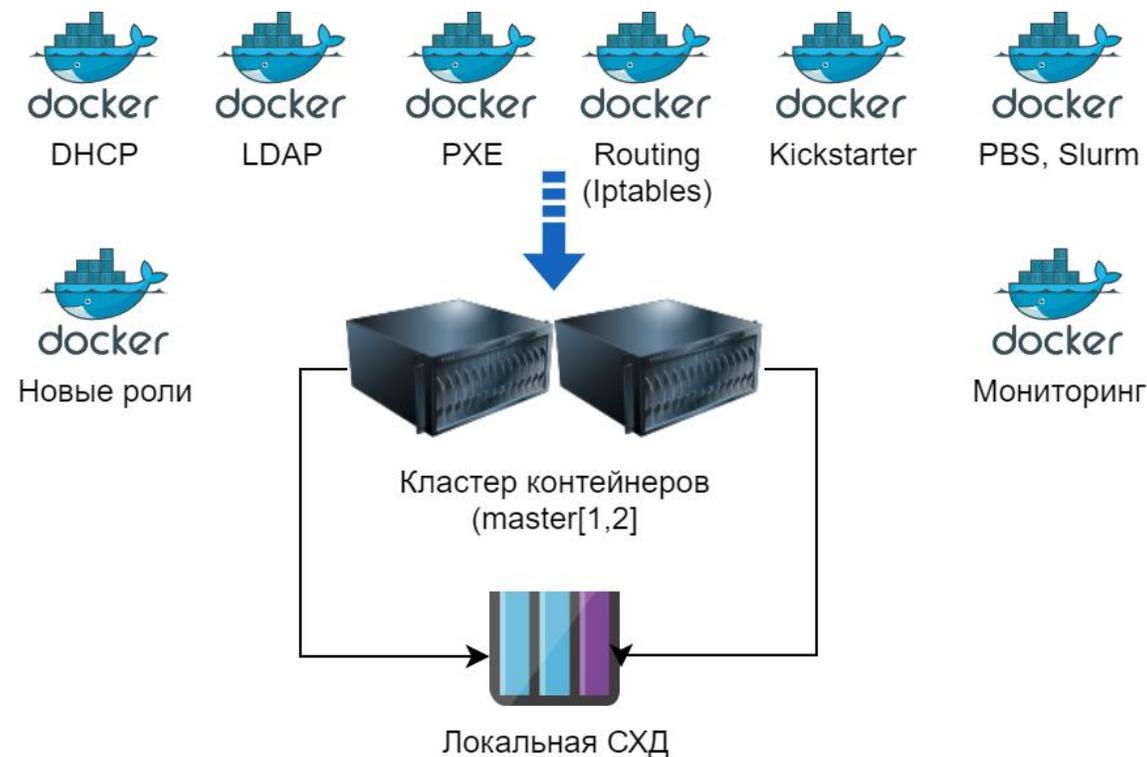


Схема контейнеризации master

**Ожидаемый результат:** остается резервирование 1+1, освобождаются 2 физических узла (резерв), появляются контейнеры с независимыми ролями и реплицированием по кластеру, облегченная миграция и перезапуск, упрощение обновления компонентов (версии контейнеров), отсутствие единой точки отказа (много небольших).

# Lawrence Berkeley National Laboratory: Singularity

## Особенности:

- Нет поддержки смены контекста (context change) → операции I/O выполняются без прослойки;
- Нет необходимости в root-правах для запуска контейнеров;
- Не допускается эскалация пользователей (user → root);
- Инкапсуляция среды;
- Контейнеры основаны на образах;
- Пользователям обеспечено взаимодействие с контейнерами;



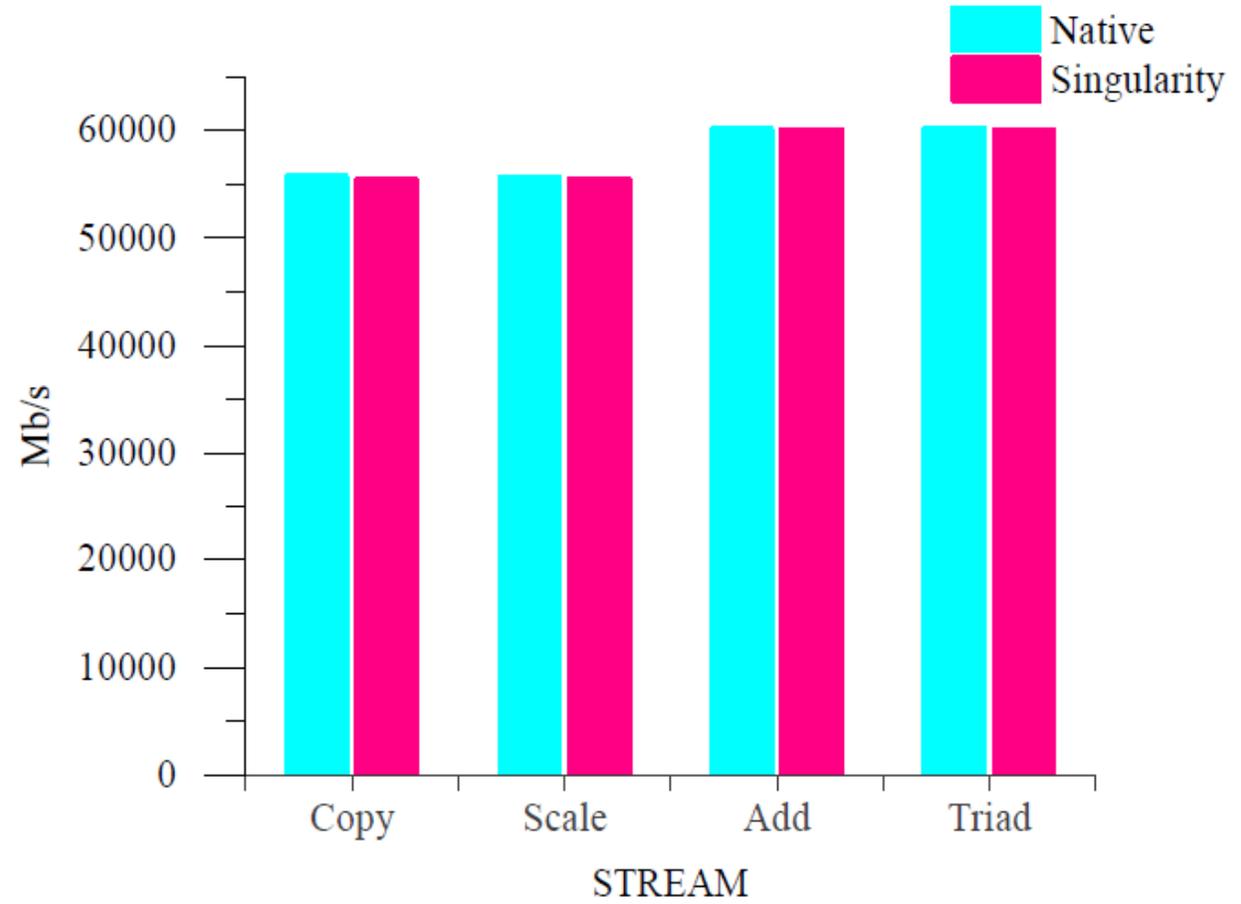
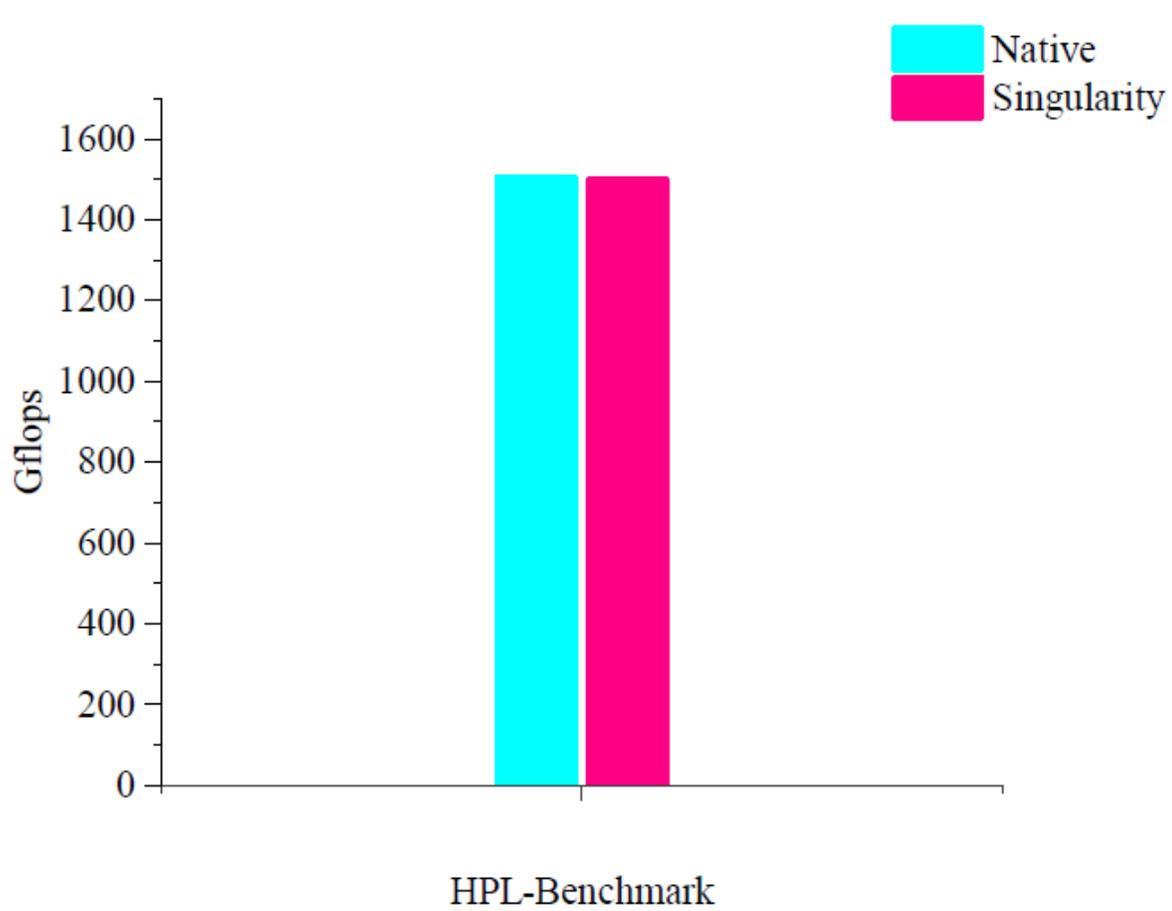
## Реализована поддержка:

- СУПЗ (Slurm, PBS Torque, SGE);
- Сетевых ФС;
- GPU;
- InfiniBand;
- MPI;
- etc.

<https://docs.sylabs.io/guides/2.6/user-guide/index.html>

**Docker VS Singularity: “can achieve near-native performance on bare metal”**

# Бенчмарки HPL, STREAM



# Контейнеризация вычислительной среды

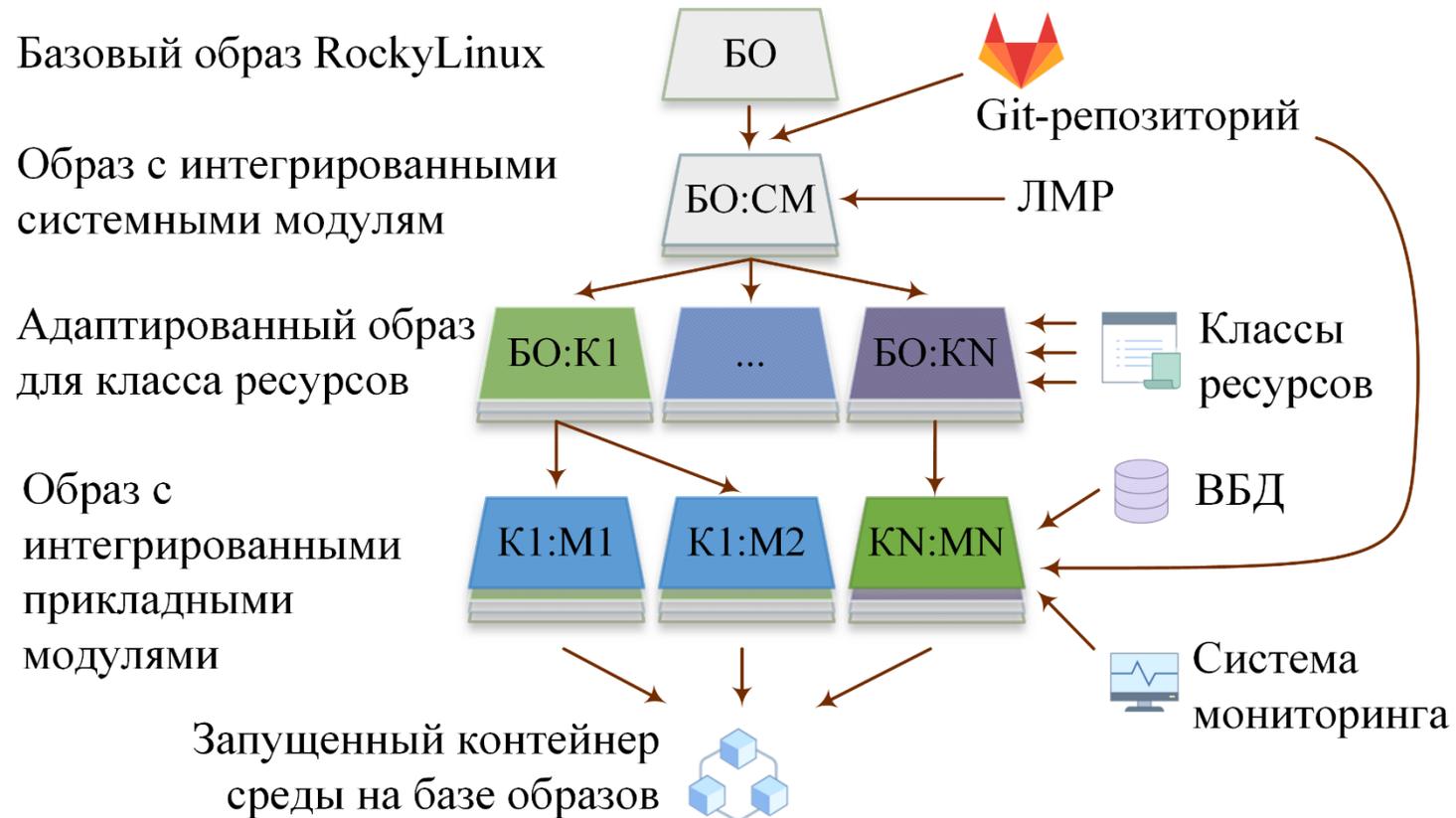
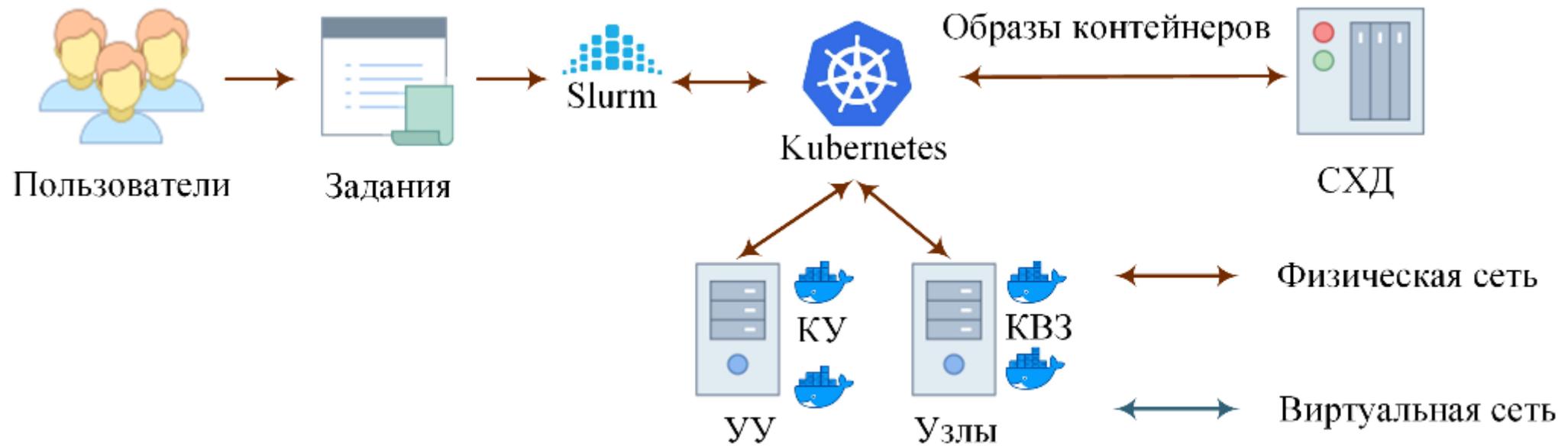


Схема подхода к контейнеризации

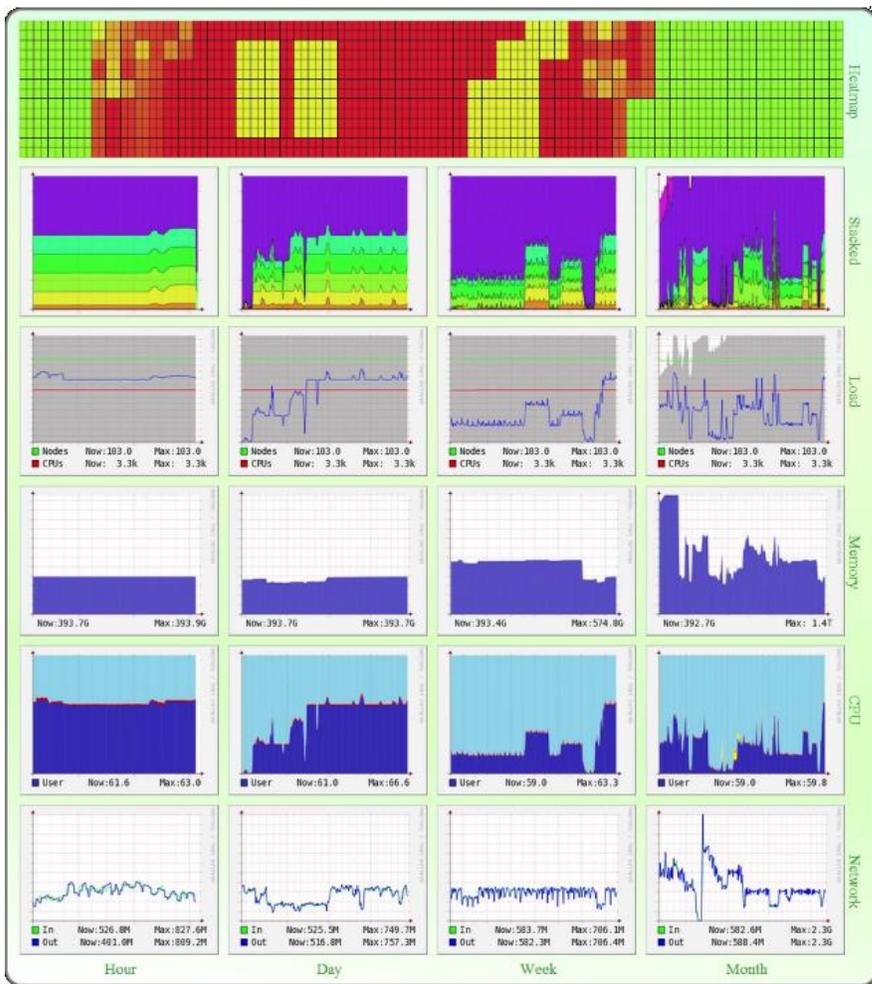
# Контейнеризация вычислительной среды



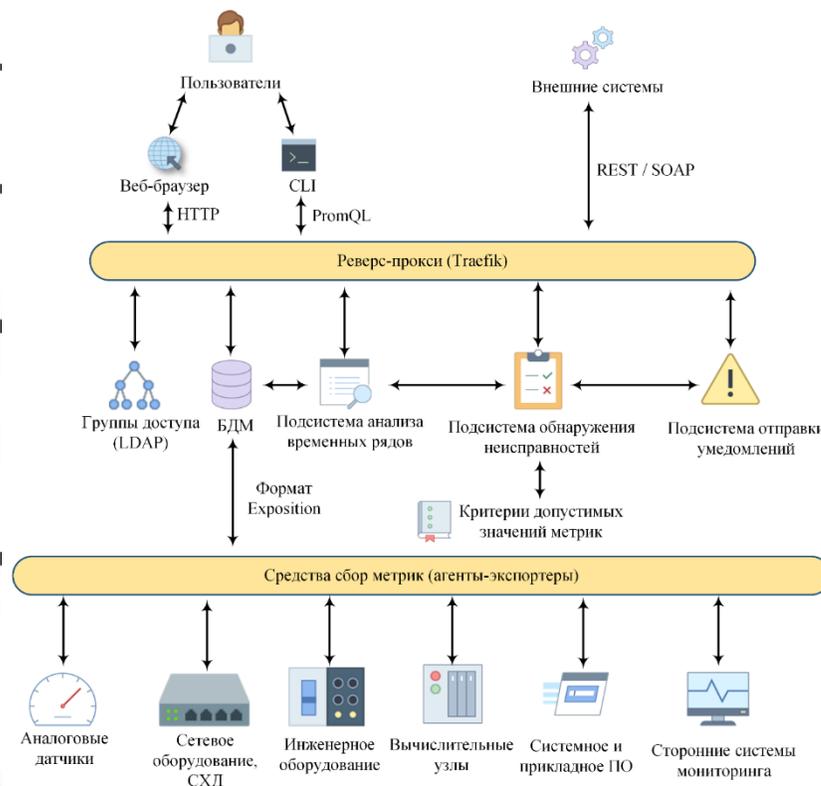
Foreman, Netbox, Ansible, Helm Charts, k8s

Схема подхода к контейнеризации вычислений

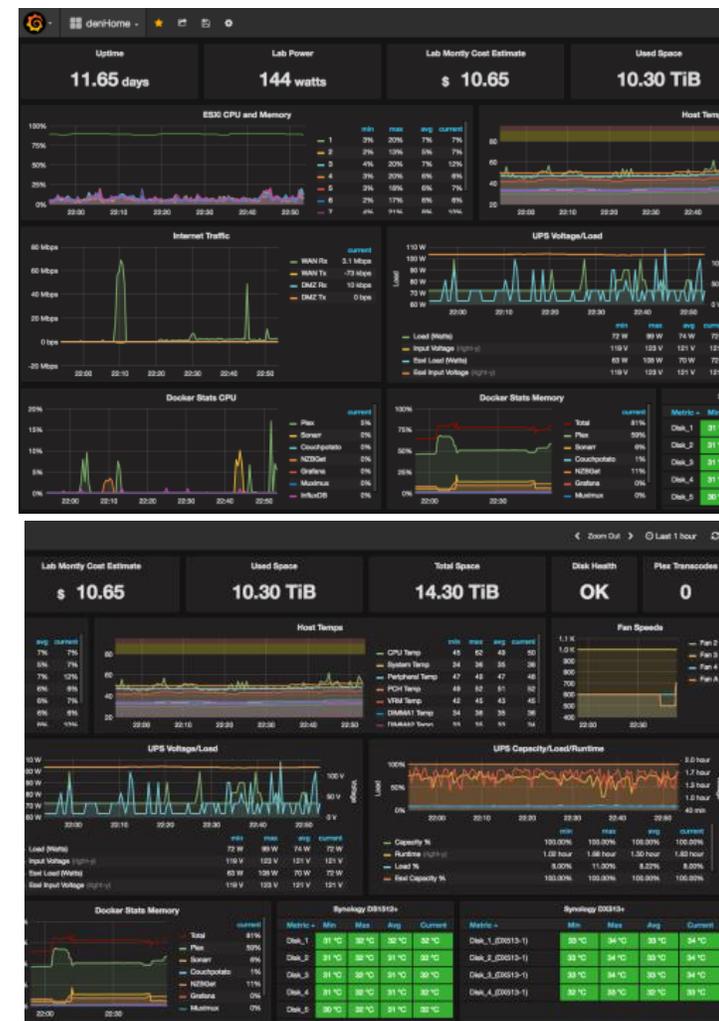
# Мониторинг: Prometheus, node\_exporter, Graphana



Действующая система мониторинга



Обобщенная схема мониторинга на базе Prometheus



Мониторинг в Graphana

# Какие преимущества применения контейнеризации?

- сохранение привычных сценариев работы с кластером;
- расширение сценариев использования кластера;
- интеграция в счетное поле облачных ресурсов;
- горизонтальное масштабирование;
- точное квотирование и выделение ресурсов;
- автоматизированная подготовка вычислительной среды;
- возможность воспроизведения эксперимента (инфраструктура как код, модули как код, workflow);
- «свежая» установка для каждого эксперимента – чистая система, окружение;
- отказоустойчивость на уровне сущностей (контейнер);
- поддержка широкого набора систем хранения (NFS, PanFS, S3);
- небольшой вес образов по сравнению с виртуальными машинами;
- снижение накладных расходов на сопровождение инфраструктуры;

Но:

- повышенные требования к квалификации системного администратора (DevOps-инженер);
- необходимость разработки новых методов планирования ресурсов;

# Благодарности

Исследование выполнено при финансовой поддержке Министерства науки и высшего образования Российской Федерации :

- Грант № 075-15-2024-533 на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (проект «Фундаментальные исследования Байкальской природной территории на основе системы взаимосвязанных базовых методов, моделей, нейронных сетей и цифровой платформы экологического мониторинга окружающей среды», 2024-2026 гг.



Спасибо за внимание!

Роман Олегович Костромин, [roman@kostromin.ru](mailto:roman@kostromin.ru)