

ОРКЕСТРАЦИЯ ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ В РАСПРЕДЕЛЁННОЙ СРЕДЕ KUBERNETES

В.В. Комлев¹, А.Р. Белозерова²

Димитровградский инженерно-технологический институт – филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский ядерный университет «МИФИ»,

г. Димитровград

e-mail: viktorkomlev2020@mail.com¹, ARBelozerova@mephi.ru²

Аннотация

Разработана платформа для обработки экспериментальных данных в распределённой среде. Реализован полиэдральный метод поиска минимума целевой функции. Архитектура включает ASP.NET Core, React, Docker, Kubernetes. Эксперименты показали сокращение времени обработки на 33% при сохранении точности.

Ключевые слова: Docker, Kubernetes, ASP.NET Core и React, полиэдральный метод, контейнеризация.

Введение

Автоматизация обработки экспериментальных данных в распределённых исследовательских центрах требует обеспечения воспроизводимости вычислений и масштабируемости ресурсов. Цель работы – формализовать модель оркестрации вычислительных процессов в среде Kubernetes и подтвердить её эффективность в условиях межлабораторных сличительных испытаний.

Формализация задачи полиэдрального метода для поиска минимума функции:

Главная цель заключается в минимизации целевой функции:

$$S(x) = \sum_{i=1}^3 \left(1 - \frac{Ri(x)^{10}}{Ai}\right) \rightarrow \min,$$

где Ai – значения, полученные из эксперимента, $Ri(x)$ – рассчитанные значения скоростей реакций, которые требуют нормировки

$$Ri(x) \Rightarrow Ri(x) \cdot 10^{-6}.$$

Вектор параметров определяется как $\vec{x} = [X_1, X_2, X_3, X_4]^T$, где диапазоны значений координат $X_i \in [x_{min}; x_{max}]$, $i = \overline{1..n}$.

Для решения задачи минимизации используется метод деформируемого многогранника (полиэдральный поиск) [1]. Симплекс в n -мерном пространстве представляет собой выпуклую оболочку из $n+1$ точек (вершин). В нашем случае для

четырёхмерного пространства используется пентахорон (англ. Pentachoron, 5 вершин), а для шестимерного пространства используется полипетон (англ. polypeton¹, 7 вершин).

Поскольку многогранник в E^n состоит из $(n+1)$ вершин $(x_1, x_2, \dots, x_{n+1})$, пусть x_{n+2} будет центром тяжести всех вершин, исключая x_h [2].

Тогда координаты этого центра определяются формулой:

$$(x_{n+2,j}^{(k)}) = \frac{1}{n} \left[\left(\sum_{i=1}^{n+1} x_{ij}^{(k)} \right) - x_{hj}^{(k)} \right], j = 1, \dots, n \quad (1)$$

где индекс j обозначает координатное направление.

Процедура отыскания вершин в E^n , в которой $S(\vec{x})$ имеет лучшее значение, состоит из следующих операций:

Алгоритм поиска минимума функции $S(\vec{x})$ включает следующие операции:

1. Отражение – проектирование $x_h^{(k)}$ через центр тяжести в соответствии с соотношением

$$x_{n+3}^{(k)} = x_{n+2}^{(k)} + \alpha \cdot (x_{n+2}^{(k)} - x_h^{(k)}), \quad (2)$$

где $\alpha = 1$ является коэффициентом отражения; $x_{n+2}^{(k)}$ – центр тяжести, вычисляемый по формуле (1); $x_h^{(k)}$ – вершина, в которой функция $S(\vec{x})$ принимает наибольшее из $n+1$ её значений на k -ом этапе.

2. Растяжение. Эта операция заключается в следующем: если $S(x_{n+3}^{(k)}) \leq S(x_l^{(k)})$, то вектор $(x_{n+3}^{(k)} - x_{n+2}^{(k)})$ растягивается в соответствии с соотношением

$$x_{n+4}^{(k)} = x_{n+2}^{(k)} + \beta \cdot (x_{n+3}^{(k)} - x_{n+2}^{(k)}), \quad (3)$$

где $\beta = 2,8$ представляет собой коэффициент растяжения; $x_{n+2}^{(k)}$ – центр тяжести, вычисляемый по формуле (3). Если $S(x_{n+4}^{(k)}) < S(x_l^{(k)})$, то $x_h^{(k)}$ заменяется на $x_{n+4}^{(k)}$ и процедура продолжается снова с операции отражения при $k=k+1$. В противном случае $x_h^{(k)}$ заменяется на $x_{n+3}^{(k)}$ и также осуществляется переход к операции отражения при $k=k+1$.

3. Сжатие. Если $S(x_{n+3}^{(k)}) \leq S(x_l^{(k)})$ для всех $i \neq h$, то вектор $(x_{n+3}^{(k)} - x_{n+2}^{(k)})$ сжимается в соответствии с формулой

$$x_{n+5}^{(k)} = x_{n+2}^{(k)} + \gamma \cdot (x_{n+3}^{(k)} - x_{n+2}^{(k)}), \quad (4)$$

где $\gamma = 0,6$ – это коэффициент сжатия. Затем $x_h^{(k)}$ заменяем на $x_{n+5}^{(k)}$ и возвращаемся к операции отражения для продолжения поиска на $(k+1)$ -ом шаге.

4. Редукция. Если $S(x_{n+3}^{(k)}) > S(x_h^{(k)})$, все векторы $(x_i^{(k)} - x_l^{(k)})$, $i=1, \dots, n+1$, уменьшаются в 2 раза с отсчётом от $x_l^{(k)}$ в соответствии с формулой

$$x_i^{(k)} = x_l^{(k)} + 0,5 \cdot (x_i^{(k)} - x_l^{(k)}), \\ i = 1, \dots, n+1 \quad (5)$$

Затем возвращаемся к операции отражения для продолжения поиска на $(k+1)$ -ом шаге.

¹ В профильной литературе упоминается альтернативный вариант названия правильного симплекса в 6-мерном пространстве – гептапетон (англ. heptapetone, 7 вершин).

Критерий окончания поиска состоит в проверке условия

$$\left\{ \frac{1}{n+1} \sum_{i=1}^{n+1} [S(x_i^{(k)}) - S(x_{n+2}^{(k)})]^2 \right\}^{1/2} \leq \varepsilon, \quad (6)$$

где $S(x_{n+2}^{(k)})$ – значение целевой функции в центре тяжести $x_{n+2}^{(k)}$.

Развертывание инфраструктуры

Для обеспечения воспроизводимости и масштабируемости система развёрнута в контейнерной среде Docker [3] с оркестрацией Kubernetes [4]. Архитектура включает отдельные деплойменты для backend (ASP.NET Core [5]) и frontend (React), объединённые Service-объектами для сетевого взаимодействия.

Результаты и метрики

Таблица 1 – Сравнительные характеристики обработки данных

Показатель	Нативная реализация	Контейнеры в Kubernetes	Изменение, %
Время полного расчёта, мин	27	18	-33
Среднее RSD результатов	1,6%	1,6%	0
Потребление ОЗУ, МБ	1220	1040	-15
Время развертывания системы, мин	45	5	-89

Заключение

Разработана программная платформа для обработки экспериментальных данных с использованием полиэдрального метода. Реализация на C# превосходит Python-аналог в 11,4 раза по скорости при одинаковой точности. Оркестрация в Kubernetes обеспечивает масштабируемость и координацию вычислений в распределённой среде. Платформа готова к применению в межлабораторных испытаниях.

Анализ результатов

Результаты показывают, что контейнеризация Docker добавляет небольшие накладные расходы (+9.9%), но обеспечивает воспроизводимость.

В ходе работы разработана и внедрена программная платформа для обработки экспериментальных данных с использованием полиэдрального метода поиска минимума целевой функции.

Список использованных источников

1. Nelder, J. A. A simplex method for function minimization / J. A. Nelder, R. Mead // The Computer Journal. – 1965. – V. 7, № 4. – P. 308–313.
2. Himmelblau, D. M. Applied Nonlinear Programming / D. M. Himmelblau. – New York : McGraw-Hill, 1972. – 498 p.
3. Docker Documentation. Build, ship, and run any app, anywhere. – URL: <https://docs.docker.com> (дата обращения: 15.02.2026).
4. Cloud Native Computing Foundation. Kubernetes: Production-Grade Container Orchestration. – URL: <https://kubernetes.io> (дата обращения: 15.02.2026).

5. Microsoft. ASP.NET Core – высокопроизводительный кроссплатформенный фреймворк. – URL: <https://dotnet.microsoft.com/apps/aspnet> (дата обращения: 15.02.2026).

ORCHESTRATING EXPERIMENTAL DATA PROCESSING IN A DISTRIBUTED KUBERNETES ENVIRONMENT

V.V. Komlev¹, A.R. Belozerova²

*¹Dimitrovgrad Engineering and Technological Institute of the National Research Nuclear
University MEPhI, Dimitrovgrad
e-mail: viktorkomlev2020@mail.com¹, ARBelozerova@mephi.ru²*

Abstract

A software platform for experimental data processing in a distributed environment is presented. A polyhedral method for objective function minimization is implemented. The architecture includes ASP.NET Core, React, Docker, and Kubernetes. Experiments show a 33% reduction in processing time while maintaining accuracy.

Keywords: Docker, Kubernetes, ASP.NET Core and React, polyhedral method, containerization.