

УДК 004.418

## РЕАЛИЗАЦИЯ ПЕРЕНОСА СОСТОЯНИЯ ИНТЕРПРЕТАТОРА PYTHON МЕЖДУ ВЫЧИСЛИТЕЛЬНЫМИ УЗЛАМИ С ПРИМЕНЕНИЕМ CRIU И СРАВНИТЕЛЬНЫЙ АНАЛИЗ

Ильин Я.Д. (Университет ИТМО)

Научный руководитель – кандидат физико-математических наук, доцент Ильин Д.В.  
(ФГБОУ ВО «ЧГУ им. И.Н. Ульянова»)

**Введение.** Чтобы разработать serverless режим для jupyter lab необходимо поддерживать возможность исполнения кода пользователей на различных вычислительных конфигурациях, с возможностью выбора конфигурации при исполнении каждой ячейки ipython ноутбука. При этом состояние интерпретатора python должно переноситься между вычислительными узлами и вычисления должны продолжаться с того места на котором они остановились до переноса состояния. При этом необходимо поддерживать баланс времени переноса состояния и размера информации необходимой для переноса. Основным способом переноса состояния python интерпретатора является подход с сериализацией объектов, например с помощью cloudpickle, такой подход имеет проблемы с переносом произвольного пользовательского кода, что является проблемой для serverless режима в jupyter lab [1][2].

**Основная часть.** В подходе с сериализацией объектов существуют проблемы с переносом произвольного пользовательского кода, для решения этой проблемы можно полностью переносить состояние контейнера, в котором запущен jupyter lab между виртуальными машинами, для этого можно использовать CRIU, который позволяет заморозить запущенный контейнер, или отдельное приложение, и сохранить его состояние на диске. Сохраненные данные можно использовать для восстановления приложения и запуска его точно в том виде, в каком оно было во время заморозки [3].

Так же, в области машинного обучения используются виртуальные машины с GPU, CRIU по умолчанию не умеет переносить состояния между ними, но в подходе с serverless, перенос происходит в тот момент, когда исполнения закончились, что позволяет решить проблему.

Необходимо сравнить подход с переносом образа и подход с переносом состояния с использованием сериализации объектов, используя метрики:

- 1) Время затрачиваемое переноса состояния.
- 2) Время на создание и восстановление состояния.
- 3) Объем информации необходимой для переноса состояния.

**Выводы.** Реализован перенос состояния интерпретатора python между вычислительными узлами с применением CRIU и проведен сравнительный анализ с альтернативным подходом.

### Список использованных источников:

1. Документация к библиотеке cloudpickle. Режим доступа: <https://github.com/cloudpipe/cloudpickle/>.
2. Официальный сайт jupyter lab. Режим доступа: <https://jupyter.org/>.
3. Официальный сайт программного обеспечения CRIU. Режим доступа: <https://criu.org/>.

Ильин Я.Д. (автор)

Ильин Д.В. (научный руководитель)