



# Распределенный статический анализ

Илья Кайсин, Академический университет

Научный руководитель — Александр Подхалюзин

# Задача

Есть код на Java, нужно узнать тип каждой переменной.

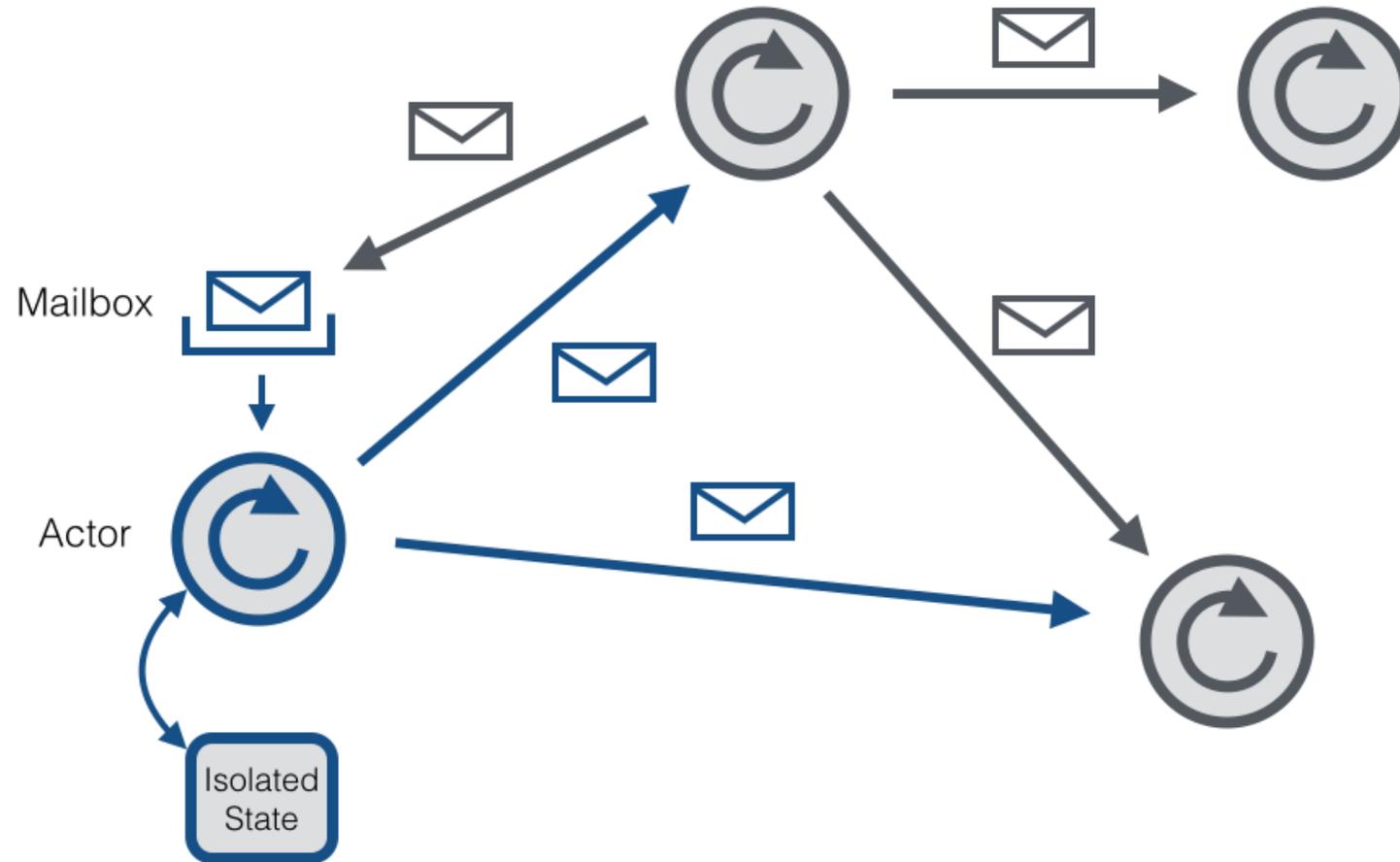
Вызываем `foo.bar()`, хотим убедиться, что у `foo` есть метод `bar`.

Вычисления производятся в несколько потоков, анализ можно раскидать по кластерам.

# Модель акторов

- ◆ Актор — независимая сущность, работающая в своем потоке.
- ◆ У акторов нет общих данных.
- ◆ Актор не может вызвать напрямую метод другого актора.
- ◆ Для коммуникации акторы отправляют сообщения.
- ◆ Актор получает сообщение, обрабатывает его, изменяет свое состояние, отправляет сообщения другим акторам.

# Модель акторов



# Что использовалось



Язык программирования. Совмещает ООП и функциональную парадигму.



Тулкит, реализующий модель акторов



Генератор парсера

# Алгоритм

- ◆ Актор создается на каждый файл, класс, метод
- ◆ Для каждой переменной определяем множество ВОЗМОЖНЫХ ТИПОВ
- ◆ Intraprocedural analysis — строится propagation граф, связывающий переменные согласно их типам  
пример:  $v1 = v2 \Rightarrow \text{Types}(v1) \supseteq \text{Types}(v2)$
- ◆ Interprocedural analysis — проталкиваем информацию через вызовы

# Что получилось

- ◆ Обход АСТ визитором
- ◆ Актор-система на уровне единиц компиляции
- ◆ Разрешение импортов
- ◆ Разрешение extends-блоков

# Что узнал

- ◆ Принципы работы IDE и компиляторов
- ◆ Парадигма акторов
- ◆ Язык Scala

# Материалы

- ◇ [github.com/demarkok/Distributed-static-analyzer](https://github.com/demarkok/Distributed-static-analyzer) – github репозиторий
- ◇ [microsoft.com/en-us/research/publication/toward-full-elasticity-distributed-static-analysis/](https://microsoft.com/en-us/research/publication/toward-full-elasticity-distributed-static-analysis/) – статья

## Вопросы?