

Acronis

Планировщик корутин

Кравченко Дмитрий

Руководитель: Коротаев Кирилл Сергеевич

Весна 2017

Мотивация

В процессе работы приложения (например, сервера) часто приходится иметь дело с вводом-выводом.

Виды IO:

1. Синхронный (C10K problem, легко писать)
2. Асинхронный (эффективный, сложно писать)

Мотивация

Синхронный

```
fs.readFileSync("file.txt")  
... // здесь окажемся только после чтения
```

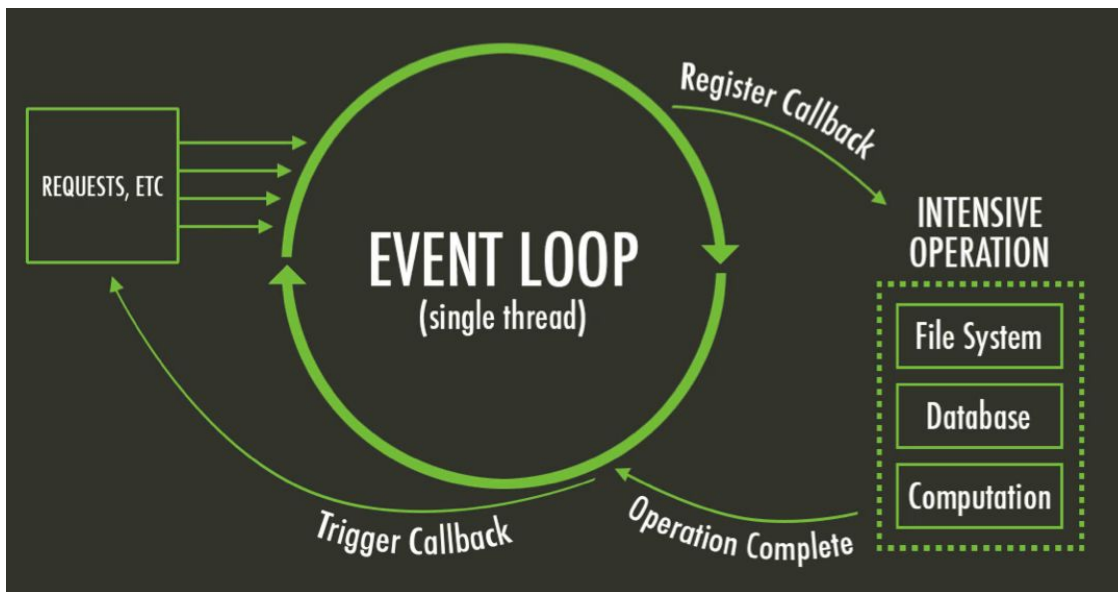
Асинхронный

```
fs.readFile("file.txt", function(err, info) {...})  
... // здесь окажемся сразу же, возможно до чтения
```

Хотим совместить два подхода и писать асинхронный код в синхронном стиле.

Мотивация

Есть глобальный event-loop, текущая реализация работает однопоточно.



Задачи

Глобальная: научиться исполнять корутины многопоточно, при этом имея load balancing + coroutine_can_block/coroutine_cant_block

Задача большая, разбили на несколько поменьше. Для начала просто научиться запускать несколько event-loop'ов с общей очередью задач.

Аналоги

libev (что-то полуживое)

libuv (Node.js) (event-loop, нет корутин)

goroutines (Go) -- примерно то, что мы хотим

Что сделано?

Начал большой рефакторинг, начал реализовывать запуск нескольких `|| event-loop`'ов, пока без оптимизаций, `load balancing`'а и т.п.

Из основных проблем: кода много, нигде никакой синхронизации + однопоточная логика + зависимость других репозиториев.

Спасибо за внимание

Вопросы?