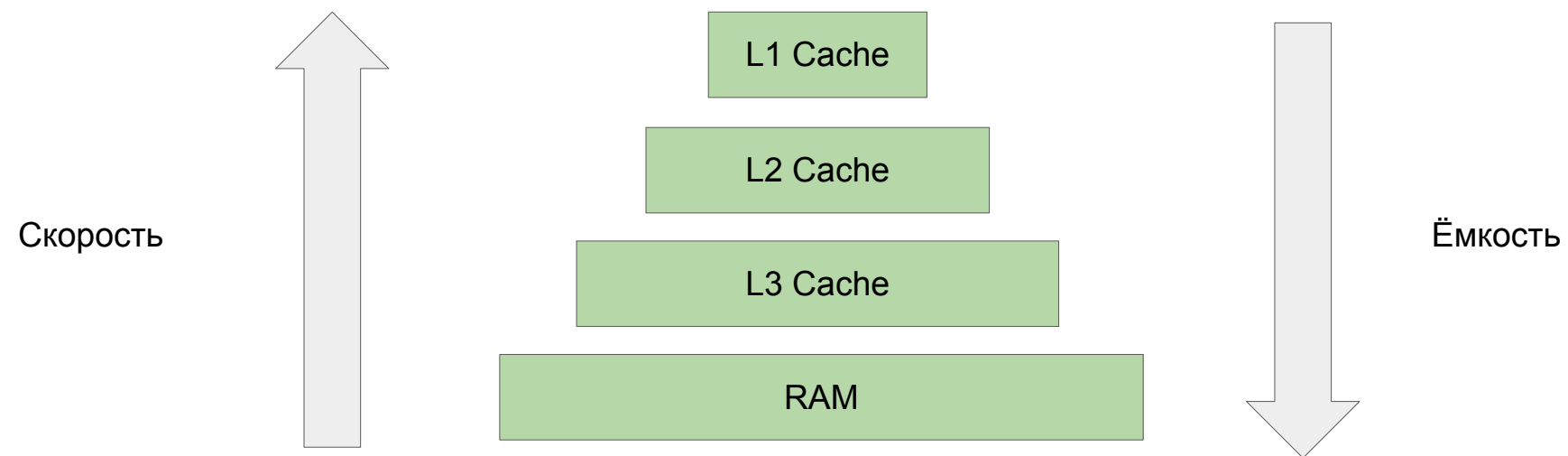


Асинхронное программирование

Утилизируем сервер

Memory

- Скорость ЦПУ растёт быстрее скорости памяти
- Чтобы это компенсировать есть кеш процессора



Context switch

- Процессор выполняет только 1 задачу одновременно
- Планировщик задач переключает их
- Для этого надо выгрузить одну и загрузить другую
- Для этого может потребоваться обращение к основной памяти



cgi-bin

Одно соединение - один процесс

Если много запросов - упрёмся в память



Multithreading

Одно соединение - один тред

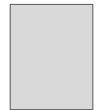
Если много запросов - упрёмся в память + дорогое переключение



А что внутри запроса?



Работа программы (операции на ЦПУ сервера)



Ожидание внешних сервисов (например, ответа от БД)

Запрос #1



Запрос #2

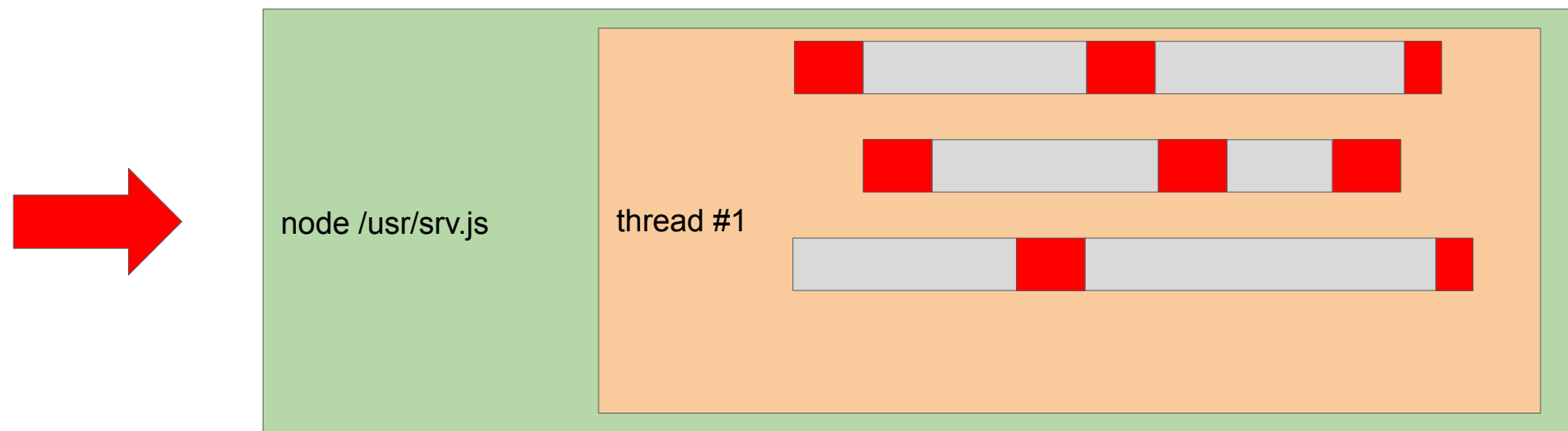


non-blocking I/O на 1 ядре

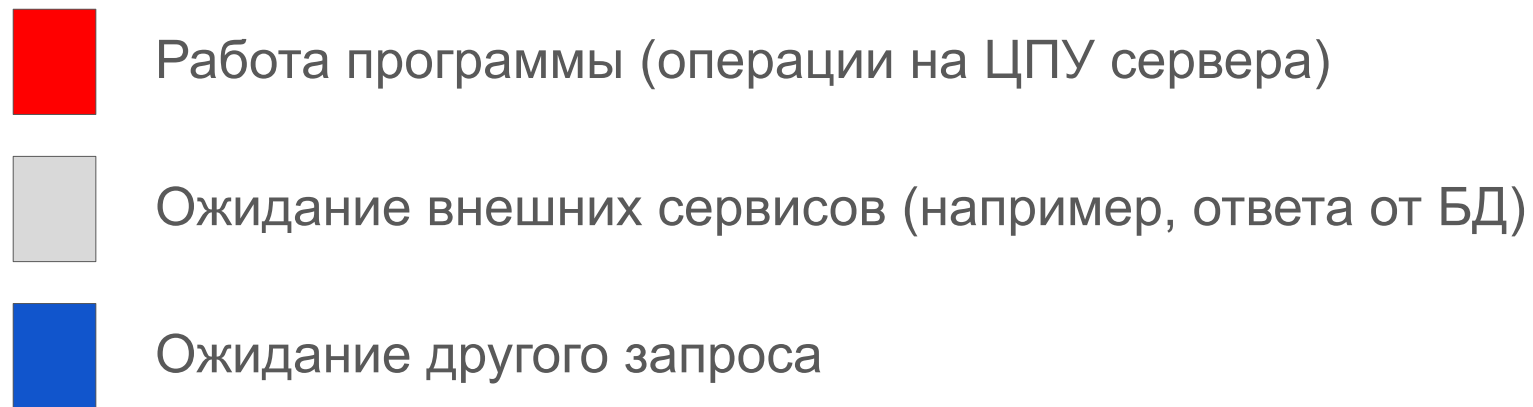
Событийная модель, кооперативная многозадачность

Одновременно работает только 1 запрос, но I/O не блокирующий

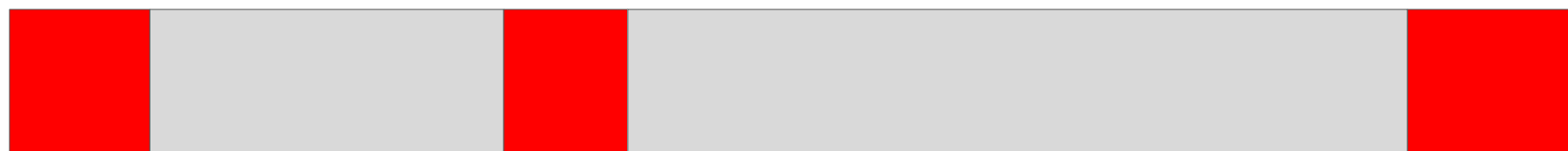
Если много запросов - упрёмся в ЦПУ



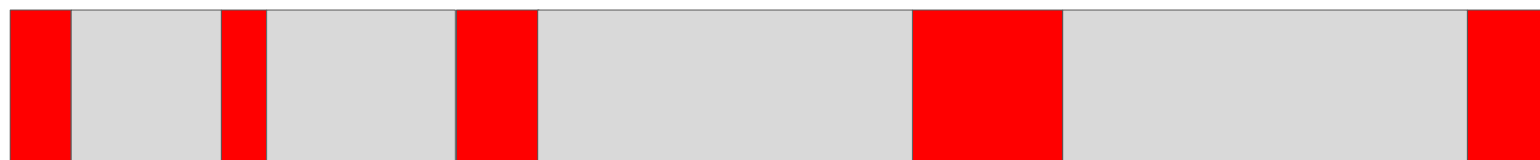
non-blocking I/O на 1 ядре



Запрос #1



Запрос #2 - ожидание



Запрос #2 - реальность

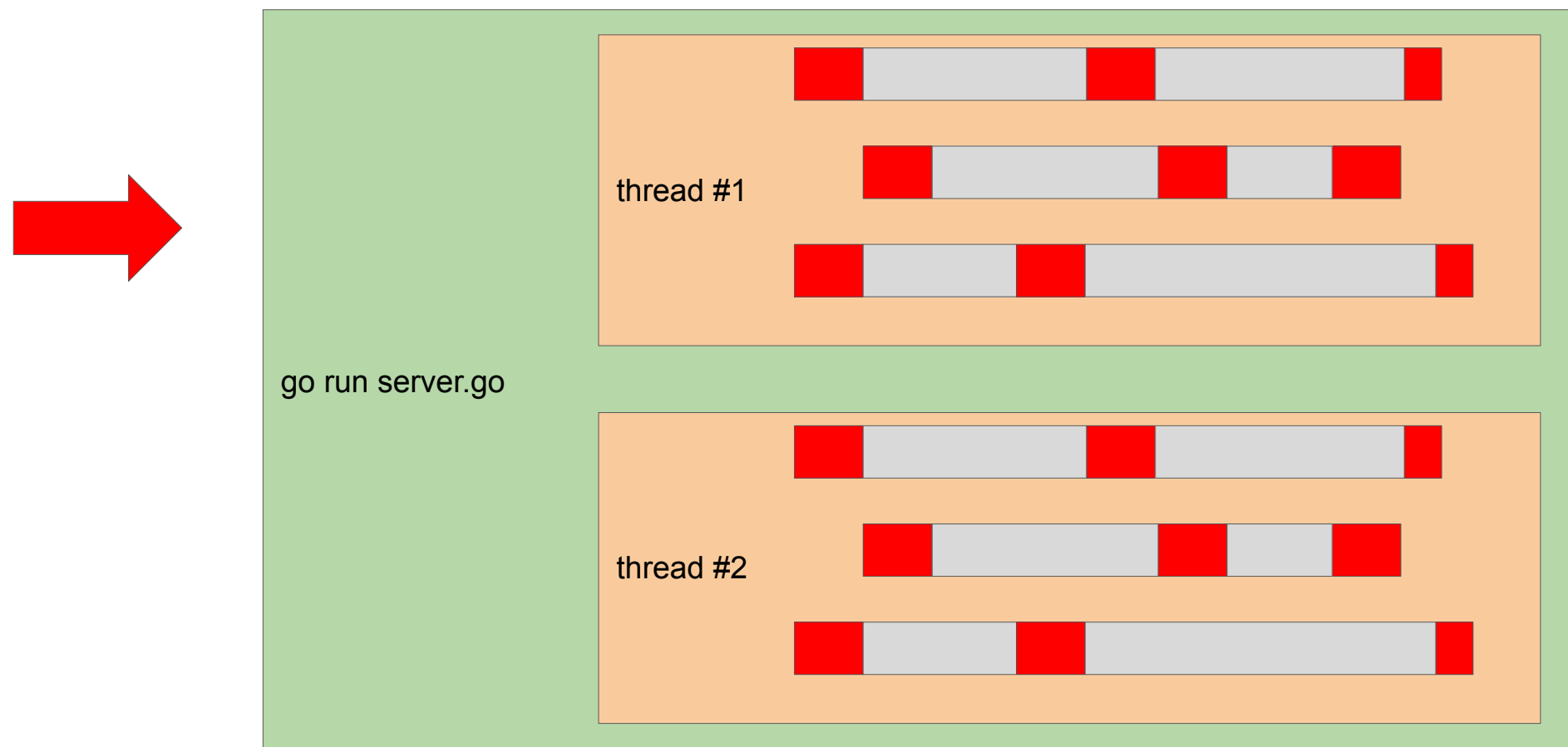


non-blocking I/O на все ядра (multithreading)

Реализовано в Golang

На основе CSP* Тони Хоара

Оперирует легковесными потоками - горутинами



- CSP - communicating sequential processes, взаимодействующие последовательные процессы