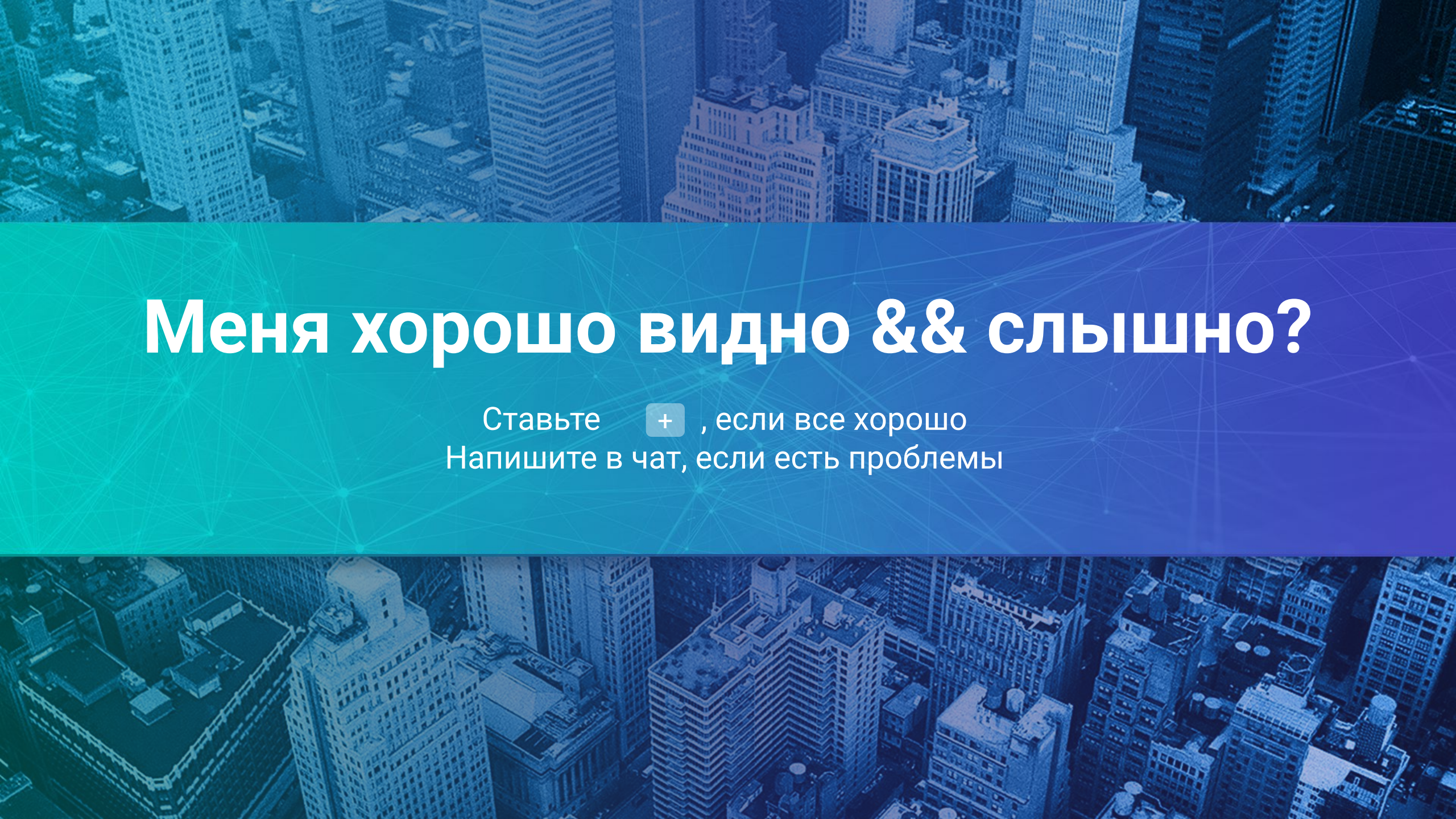

OTUS
ОНЛАЙН-ОБРАЗОВАНИЕ
Онлайн-образование

Проверить, идет ли запись!



Меня хорошо видно && слышно?

Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

Модуль 2. Стандартные библиотеки и практики
Тема 9. Планировщик

Елена Граховац

elena@grahovac.me
twitter.com/webdeva

Правила вебинара

Активно участвуем

Задаем вопрос в чат или голосом

Off-topic обсуждаем в Slack #go-2019-08 или #general

Вопросы вижу в чате, могу ответить не сразу

Цели вебинара | После занятия вы сможете

1. Понять некоторые детали работы горутин
2. Писать производительные программы с учетом особенностей работы планировщика
3. Подготовиться к вопросу “Как работает планировщик в Go” на собеседовании :)

Планировщик

Зачем нужен планировщик?

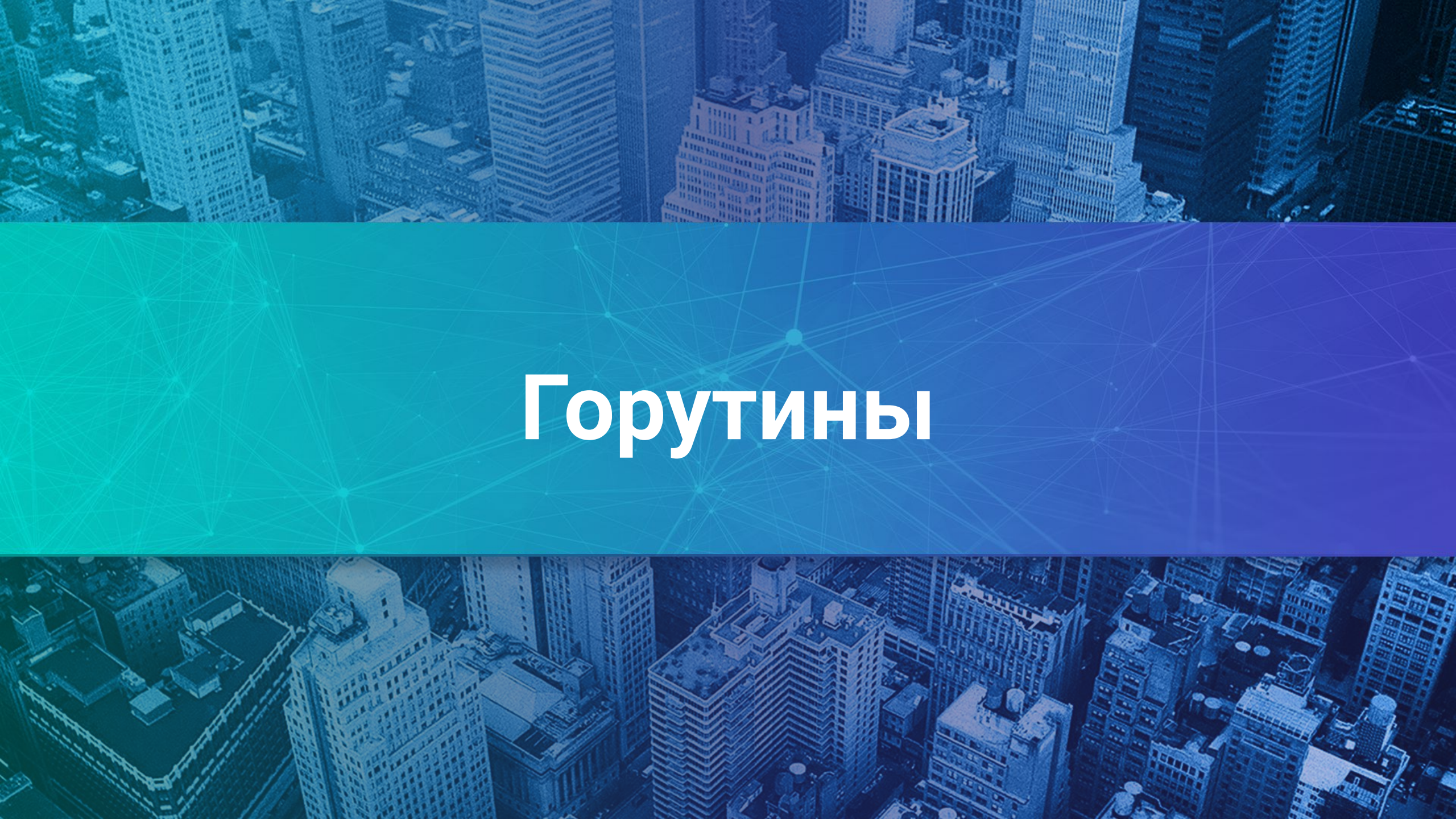
- Напишите, пожалуйста, в чат
- Зачем нужен планировщик?
- Что он делает?

Какие бывают планировщики?

- Планировщик в операционной системе
  - Управляет тем, как работают процессы
- Планировщик контейнеров
  - Управляет тем, на каких машинах создавать контейнеры
  - Да-да, Kubernetes - это пример такого планировщика :)

Планировщик Go

- Распределяет горутины на несколько тредов ОС, которые могут быть запущены на одном или нескольких процессорах
- Work stealing - свободный процессор ищет потоки других процессоров и “крадет” их - http://supertech.csail.mit.edu/papers/steal.pdf
- m:n-планирование: как выполнить m горутин на n тредах (потоках) ОС
- Планировщик вызывается во время выполнения некоторых конструкций (например: блокировка горутины операциями с каналами или time.Sleep)



Горутины

Абстракции для понимания планировщика

- **G** - горутина
- **M** - “machine” - тред (поток) ОС
- **P** - “processor” - то, что выполняет наш Go-код

Когда говорим про N:M планирование --

***N** горутин (G) выполняется на **M** потоков (M)*

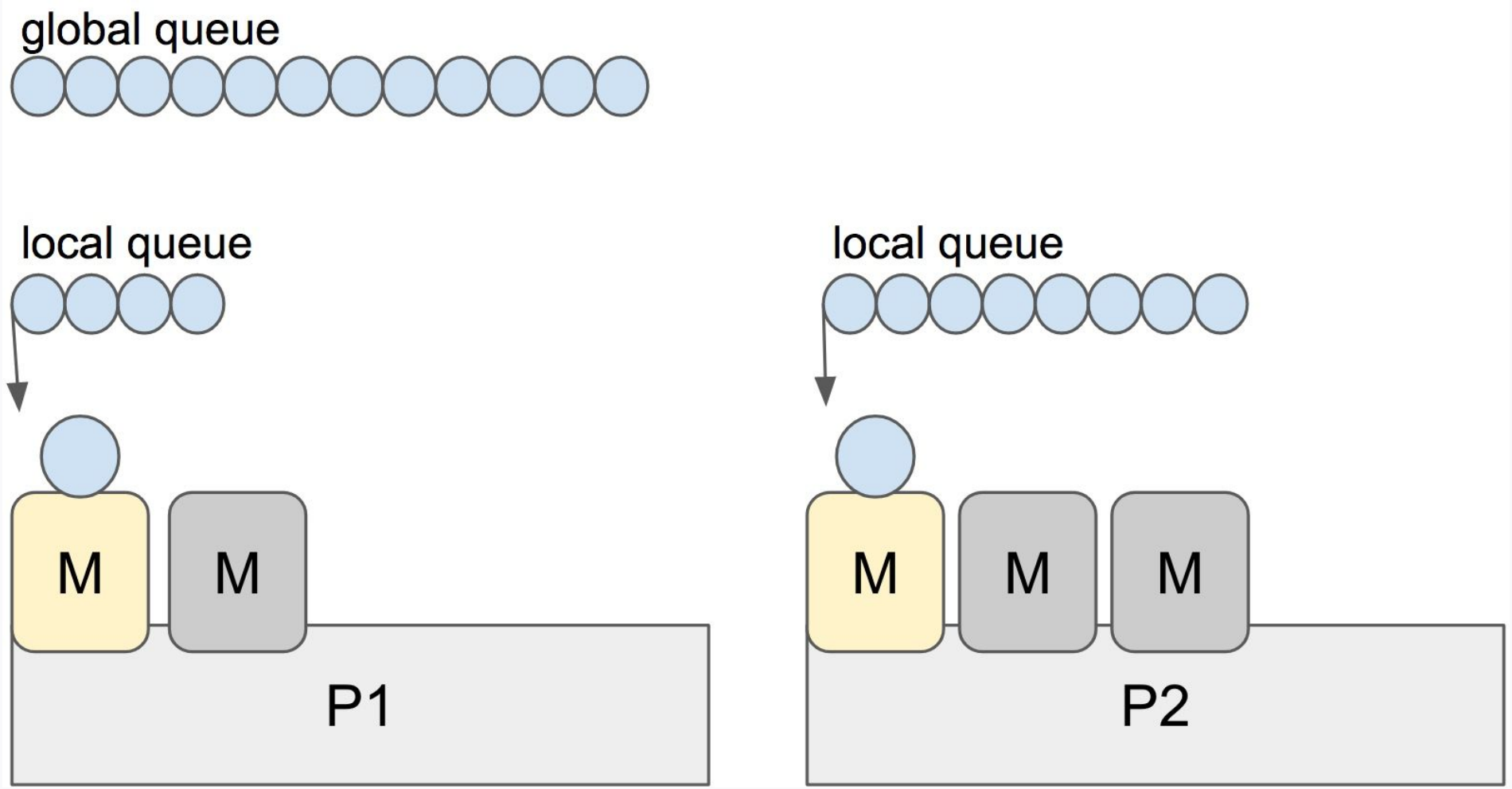
Где здесь GOMAXPROCS?

*Это количество **P**!*

Возможные состояния горутин

- Запущена в данный момент
- Готова к запуску (ждет своей очереди)
- Не готова к запуску (например, заблокирована, ждет системного вызова)

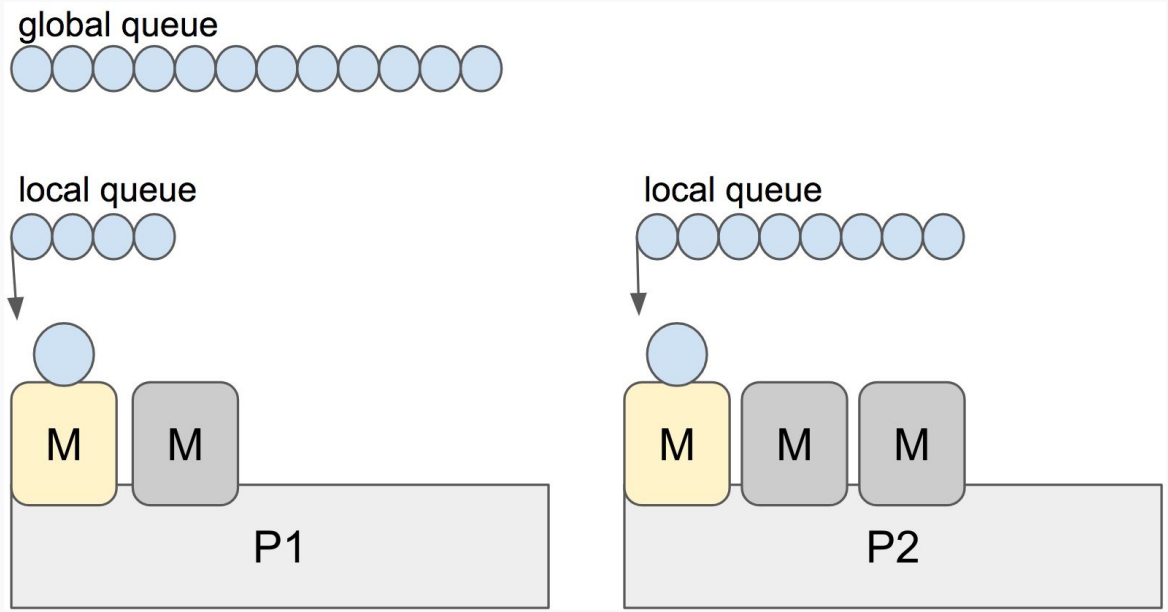
Локальные и глобальные очереди



Картинка отсюда: https://rakyll.org/scheduler/

Свойства планировщика

- Глобальная и локальная очереди
- Каждый M должен быть назначен некоторому P
- P может быть заблокирован или ждать системный вызов, тогда у P может не быть M
- Только один M может исполняться на каждом P в текущий момент
- Если нужно, планировщик может создавать дополнительные M



Цикл работы планировщика

- https://github.com/golang/go/blob/master/src/runtime/proc.go#L2468
- Время от времени (1/61) планировщик берет горутины из глобальной очереди
- Иначе - из локальной очереди текущего P
- Если в локальной очереди ничего нет, ищет и “крадет” горутины у других P

Переключение горутин

- В каком случае горутина может быть “переключена” - т.е. или определена как заблокированная, или добавлена в конец очереди на выполнение?
  - вызов runtime.Gosched()
  - операции с сетью (чтение/запись)
  - системные вызовы
  - собственно, блокировки (мьютексы, каналы)
  - некоторые аллокации, например morestack()

Переключение горутин

- Когда создается новая горутина или разблокируется существующая, она добавляется в конец очереди текущего P

Можно ли привязать горутину к конкретному треду?

- В общем случае при переключении горутины разные ее части могут выполняться на разных тредах
- Бывает, что нужно привязать горутину к конкретному треду
- Зачем?
  - Вставки на C, графические фреймворки
- Как?
  - https://github.com/golang/go/wiki/LockOSThread

Что смотреть в коде рантайма Go

- Планировщик: https://github.com/golang/go/blob/master/src/runtime/proc.go
- Еще полезное про статусы: https://github.com/golang/go/blob/master/src/runtime/runtime2.go
- Кусок про таймеры: https://github.com/golang/go/blob/master/src/runtime/time.go

Итоги занятия

Узнали что-то новое? Что кажется особенно полезным?
Напишите в чат!

Следующий вебинар

Тема:

Низкоуровневые протоколы

Ссылка на вебинар будет в ЛК за 15 минут

Материалы к занятию
в ЛК — можно изучать

Обязательный
материал обозначен
красной лентой

Список материалов для изучения: посмотреть

- Планировщик шаг за шагом: https://youtu.be/-K11rY57K7k
- Сага о планировщике: https://youtu.be/YHRO5WQGh0k
- Доклад про планировщик от JBD: https://youtu.be/Yx6FBsGNOp4
- Еще один интересный доклад (на русском!):

  https://youtu.be/Gy6XEYWYht8

Список материалов для изучения: почитать

- Design doc: https://docs.google.com/document/d/1TTj4T2JO42uD5ID9e89oa0sLKhJYD0Y_kqxDv3I3XMw/edit
- От ArdanLabs: https://www.ardanlabs.com/blog/2018/08/scheduling-in-go-part2.html
- Статья про планировщик от JBD: https://rakyll.org/scheduler/
- Пример про использование трейсера: https://blog.gopheracademy.com/advent-2017/go-execution-tracer/
- Сборник лучших статей по конкурентности: https://github.com/golang/go/wiki/LearnConcurrency
- Книга про конкруентность в Go: https://www.oreilly.com/library/view/concurrency-in-go/9781491941294/

Эти слайды доступны сразу

https://clck.ru/JjyJ3


Заполните, пожалуйста,
опрос о занятии:
https://otus.ru/polls/4908/

Спасибо за внимание!
Приходите на следующие вебинары

Елена Граховац

elena@grahovac.me
twitter.com/webdeva