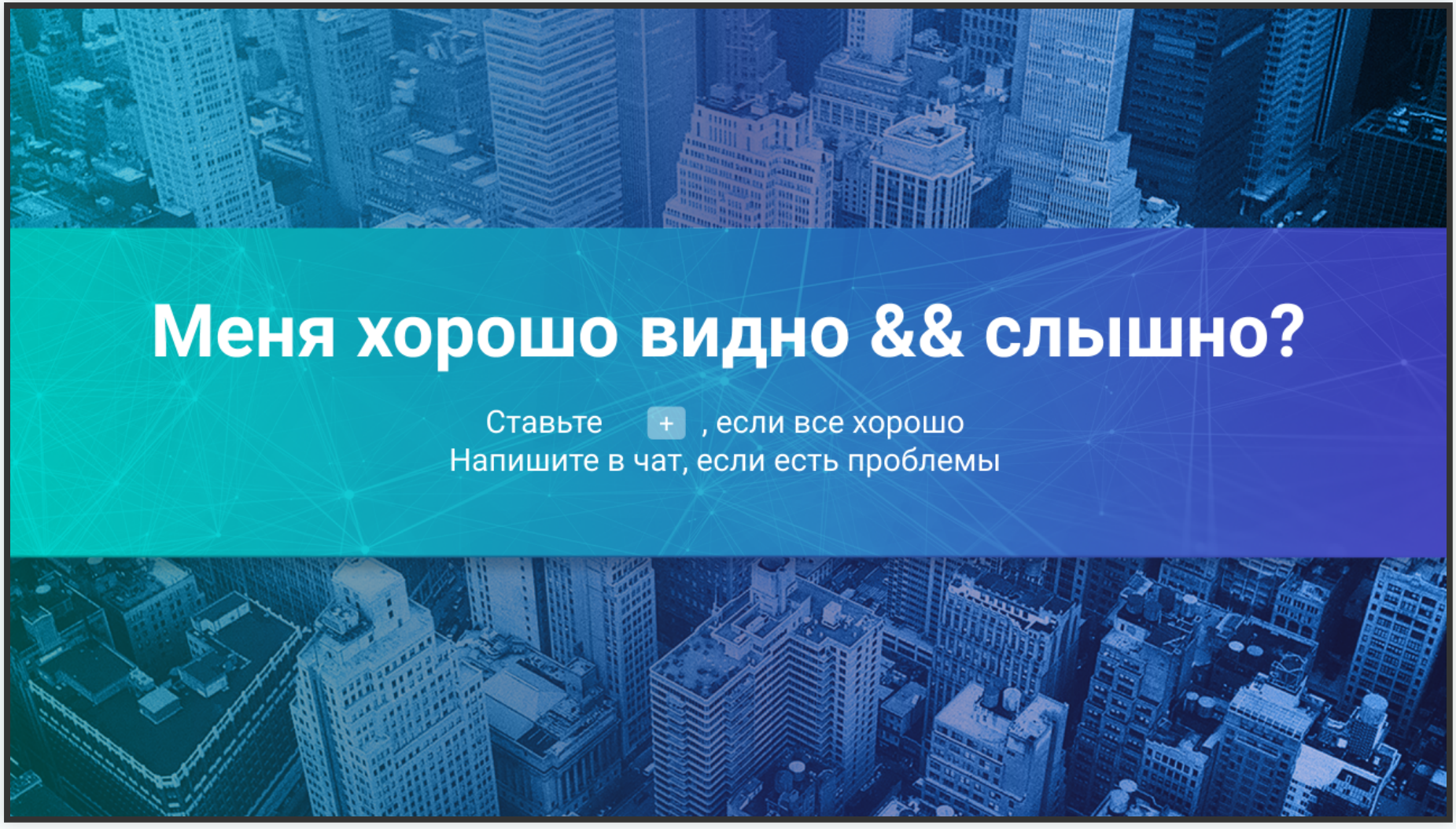

OTUS
ОНЛАЙН-ОБРАЗОВАНИЕ
Онлайн-образование


Меня хорошо видно && слышно?
Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

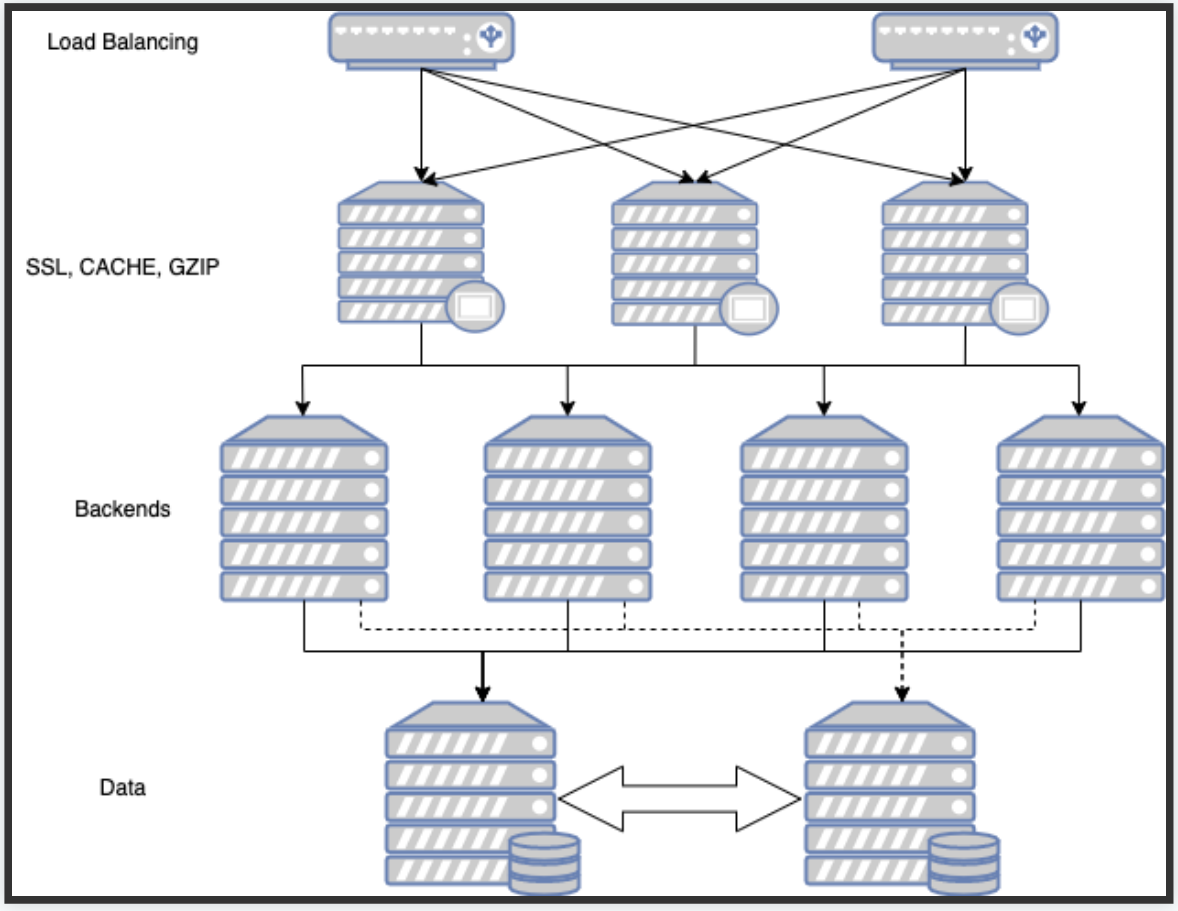
НЕ ЗАБЫТЬ ВКЛЮЧИТЬ ЗАПИСЬ!!!

Nginx. High Load.

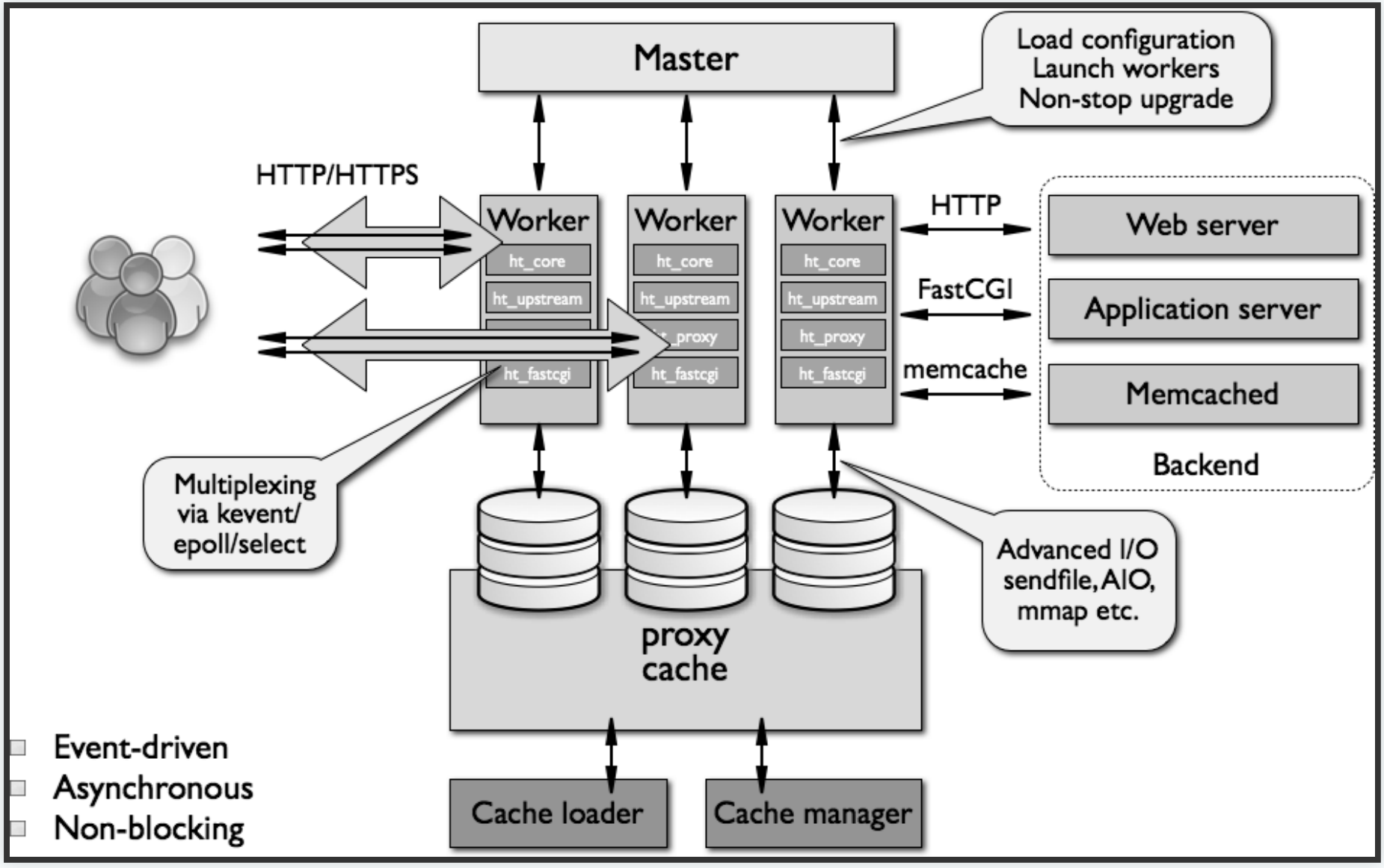
Сегодня мы рассмотрим

- принцип работы сервера nginx (epoll)
- нагрузочное тестирование через yandex tank
- оптимизация конфигурации
  - кол-во воркеров
  - настройка таймоутов
- http2
- limits
- оптимизация настроек sysctl

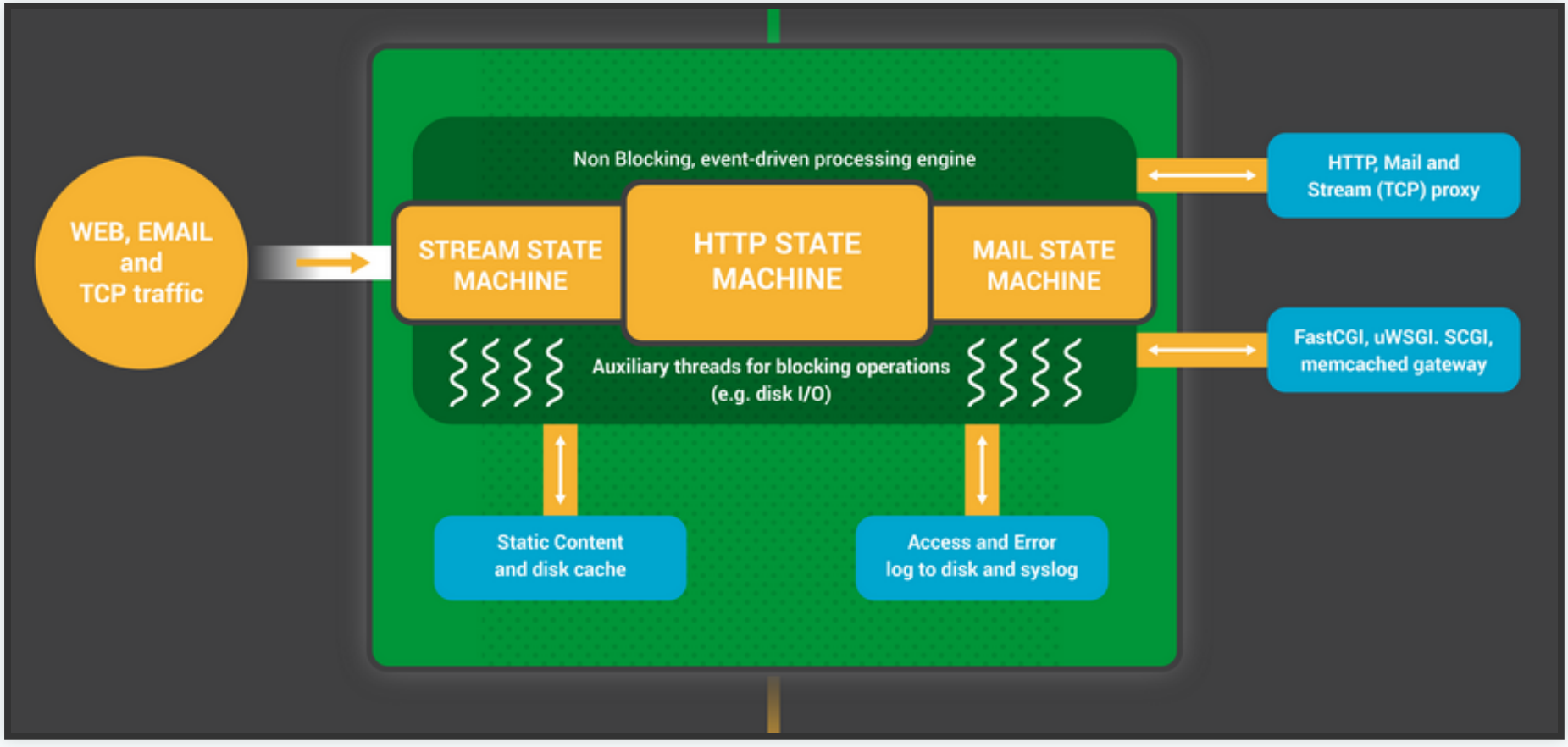
Пример балансировки



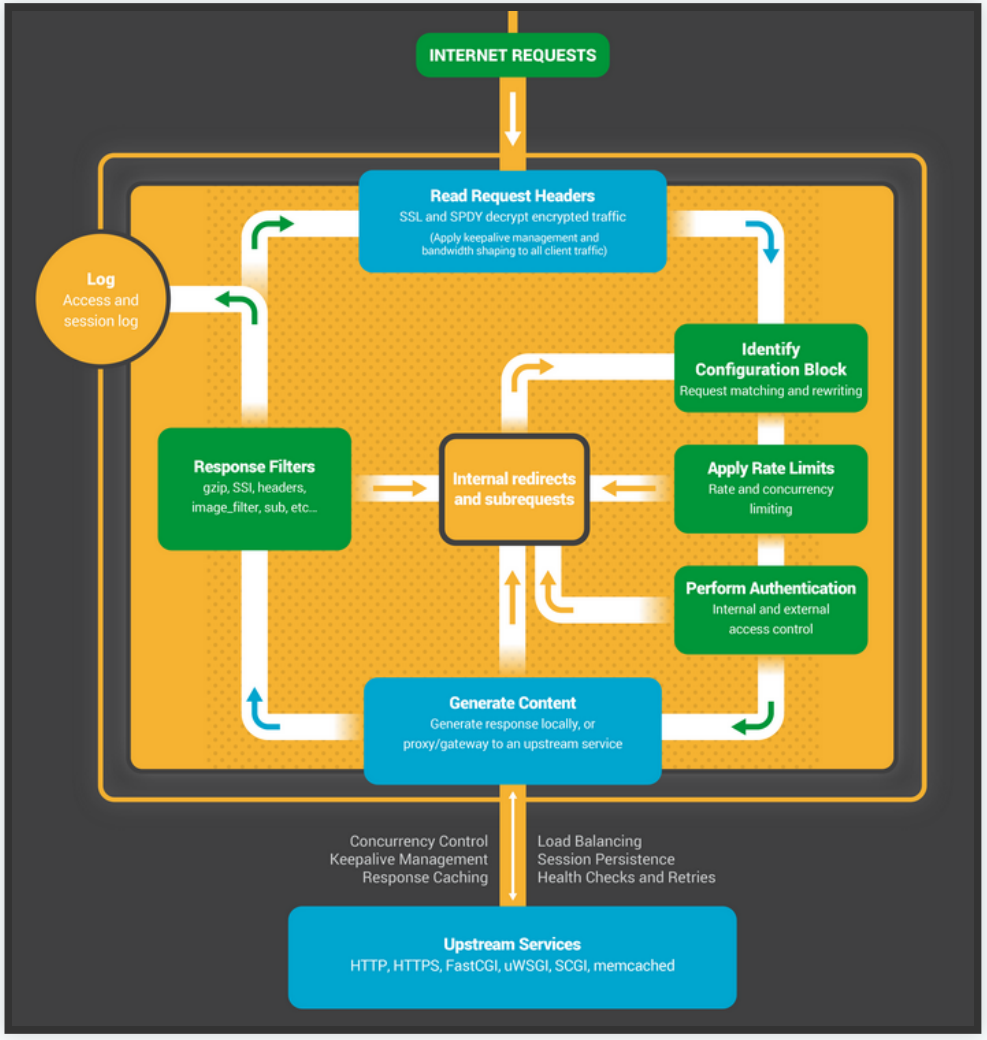
Архитектура



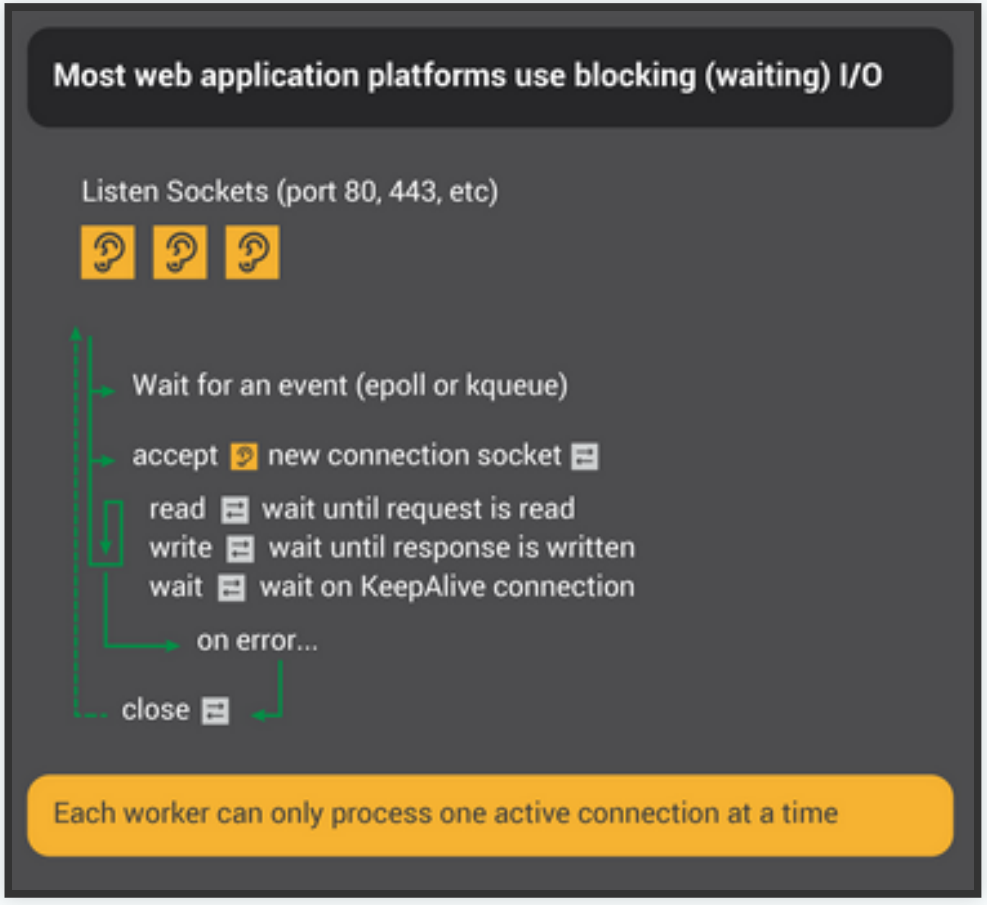
Внутри процесса вокрера



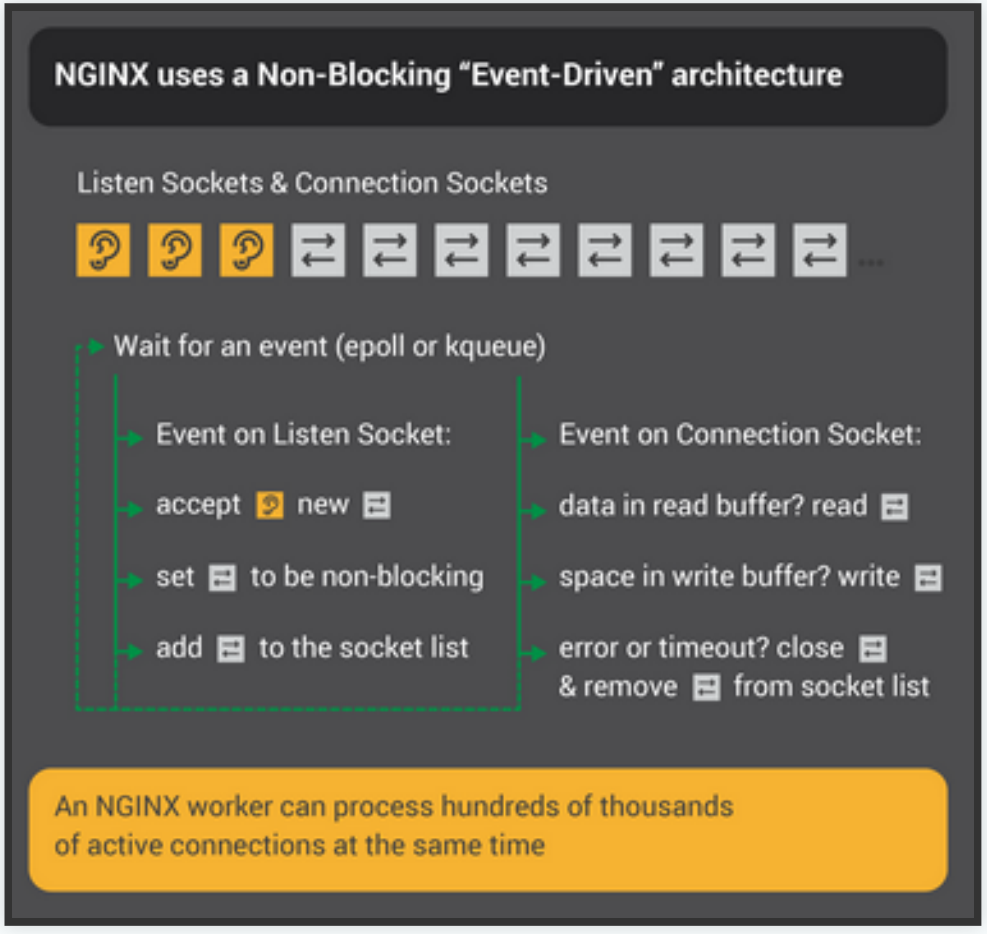
Обработка реквеста



Блокирующий конечный автомат



Неблокирующий конечный автомат NGINX



Методы обработки соединений

- nginx поддерживает различные методы обработки соединений.
- Наличие того или иного метода зависит от используемой платформы.
- Если на платформе доступно сразу несколько методов, nginx обычно сам выбирает наиболее эффективный метод.
- можно явно выбрать метод обработки соединений с помощью директивы use.

Методы обработки соединений

- select - стандартный метод.
- poll - стандартный метод.
- epoll - эффективный метод, используемый в Linux 2.6+.
- kqueue - FreeBSD 4.1+, OpenBSD 2.9+, NetBSD 2.0 и macOS.
- /dev/poll — эффективный метод, используемый в Solaris 7 11/99+, HP/UX 11.22+ (eventport), IRIX 6.5.15+ и Tru64 UNIX 5.1A+.
- eventport — event ports, метод, используемый в Solaris 10+ (из-за имеющихся проблем вместо него рекомендуется использовать метод /dev/poll).

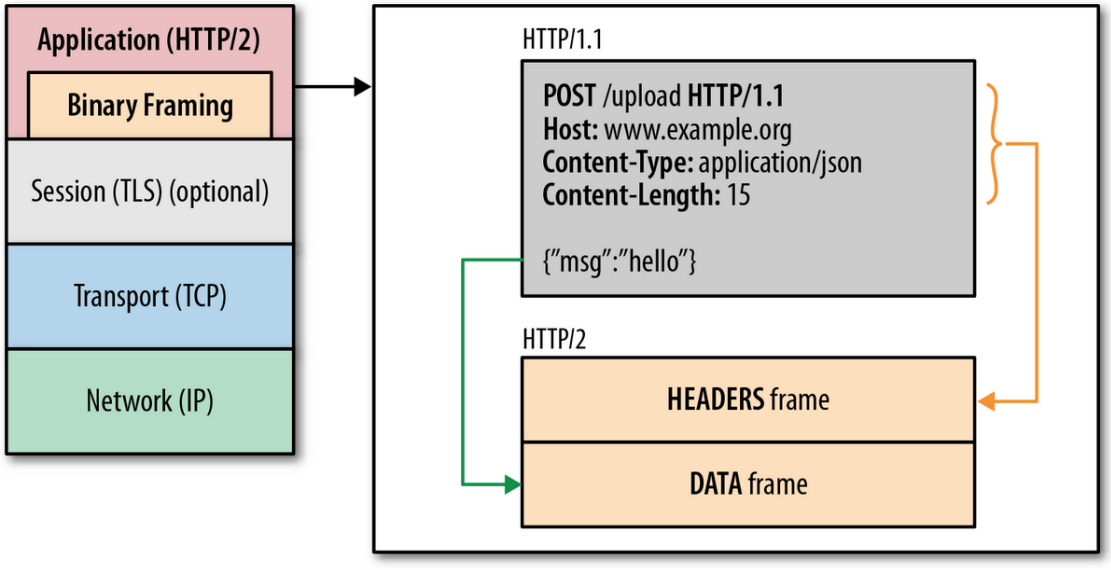
HTTP2

```
server {
    # Ensure that HTTP/2 is enabled for the server
    listen 443 ssl http2;
    ssl_certificate ssl/certificate.pem;
    ssl_certificate_key ssl/key.pem;
    root /var/www/html;
    # whenever a client requests demo.html, also push
    # /style.css, /image1.jpg and /image2.jpg
    location = /demo.html {
        http2_push /style.css;
        http2_push /image1.jpg;
        http2_push /image2.jpg;
    }
}
```

HTTP2

HTTP/2 in one slide…

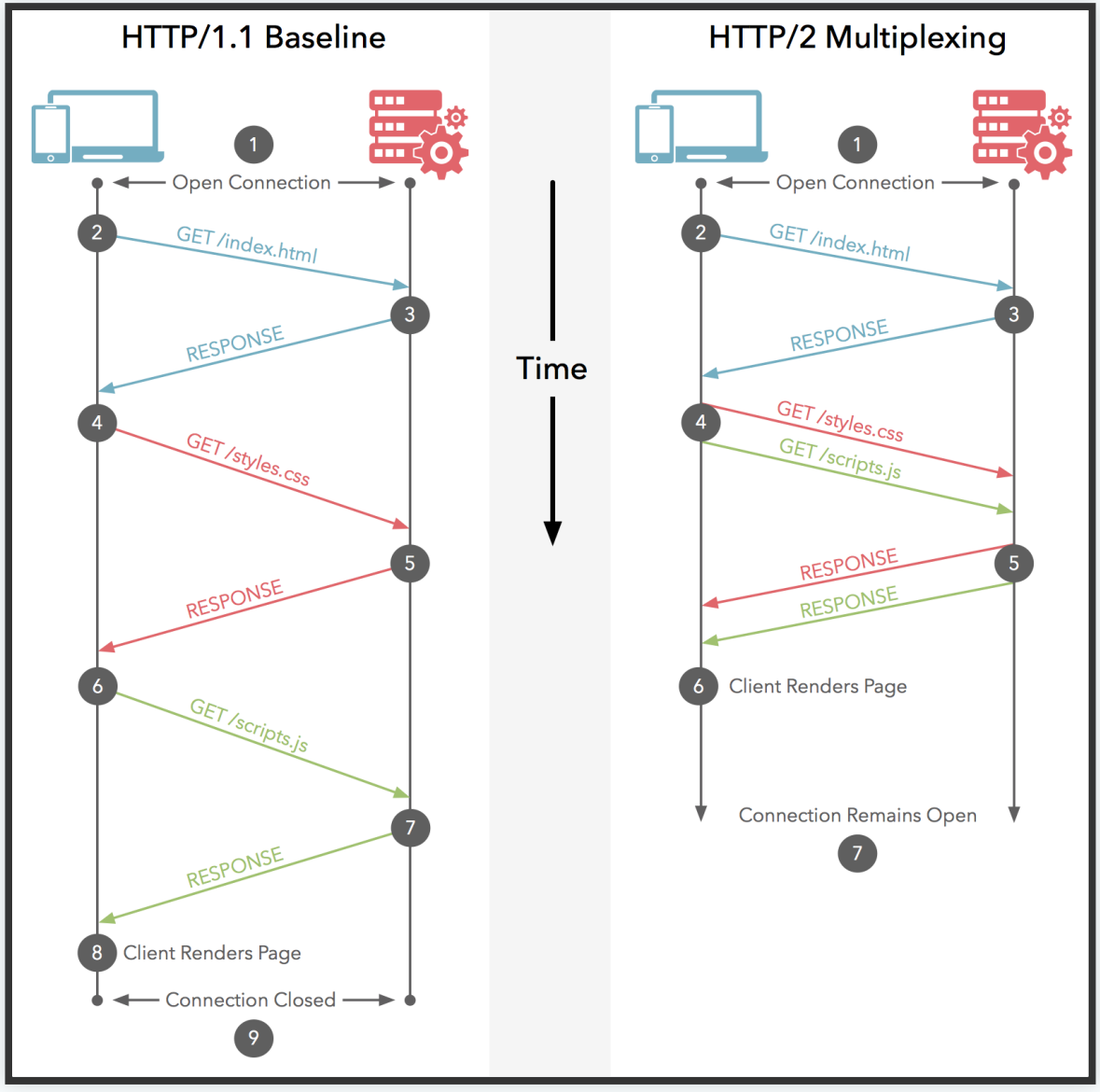
1. **One TCP connection**
2. **Request → Stream**
   - Streams are multiplexed
   - Streams are prioritized
3. **Binary framing layer**
   - Prioritization
   - Flow control
   - Server push
4. **Header compression (HPACK)**



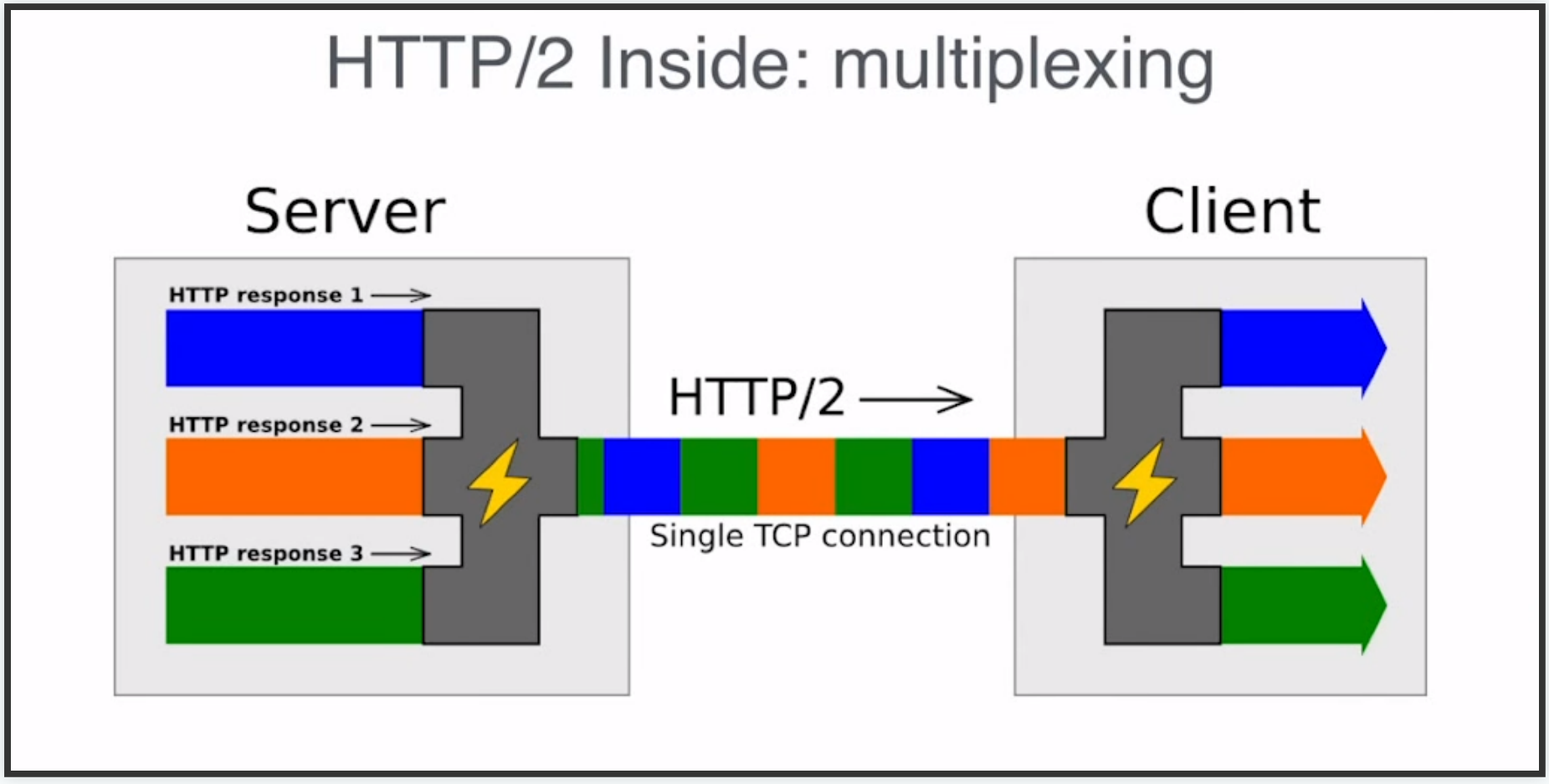
Преимущества HTTP2

- бинарный протокол
- Мультиплексирование
- Компрессия заголовков
- Приоритизация
- push с сервера на клиент

HTTP2 - мультиплексирование



HTTP2 - мультиплексирование



HTTP2 - мультиплексирование

HTTP/2 Inside: binary

HTTP/2.0 request:

```
00 00 9D 01 25 00 00 00  01 00 00 00 00 B6 41 8A    ..%  .    .A.
90 B4 9D 7A A6 35 5E 57  21 E9 82 00 84 B9 58 D3   ...z.5^W!.. ..X.
3F 85 61 09 1A 6D 47 87  53 03 2A 2F 2A 50 8E 9B   ?.a..mG.S.*/*P..
D9 AB FA 52 42 CB 40 D2  5F A5 11 21 27 51 8B 2D   ...RB.@._..!'Q.-
4B 70 DD F4 5A BE FB 40  05 DE 7A DA D0 7F 66 A2   Kp..Z..@..z...f.
81 B0 DA E0 53 FA D0 32  1A A4 9D 13 FD A9 92 A4   ....S..2........
96 85 34 0C 8A 6A DC A7  E2 81 04 41 04 4D FF 6A   ..4..j.....A.M.j
43 5D 74 17 91 63 CC 64  B0 DB 2E AE CB 8A 7F 59   C]t..c.d.......Y
B1 EF D1 9F E9 4A 0D D4  AA 62 29 3A 9F FB 52 F4   .....J...b):..R.
F6 1E 92 B0 D3 AB 81 71  36 17 97 02 9B 87 28 EC   .......q6.....(.
33 0D B2 EA EC B9
```

HTTP/1.1 request:

```
GET / HTTP/1.1
Host: demo.nginx.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Chrome/47.0.2518.0
```

Limits

- ulimit - управление ресурсами пользователя
- prlimit - управление ресурсами процесса
- help ulimit
- ulimit -a
- man prlimit
- cat /etc/security/limits.conf
- `# <domain> <type> <item> <value>`

Управление лимитами

- пользователь может менять свои soft лимиты максимум до hard значени
- hard значения может менять только root

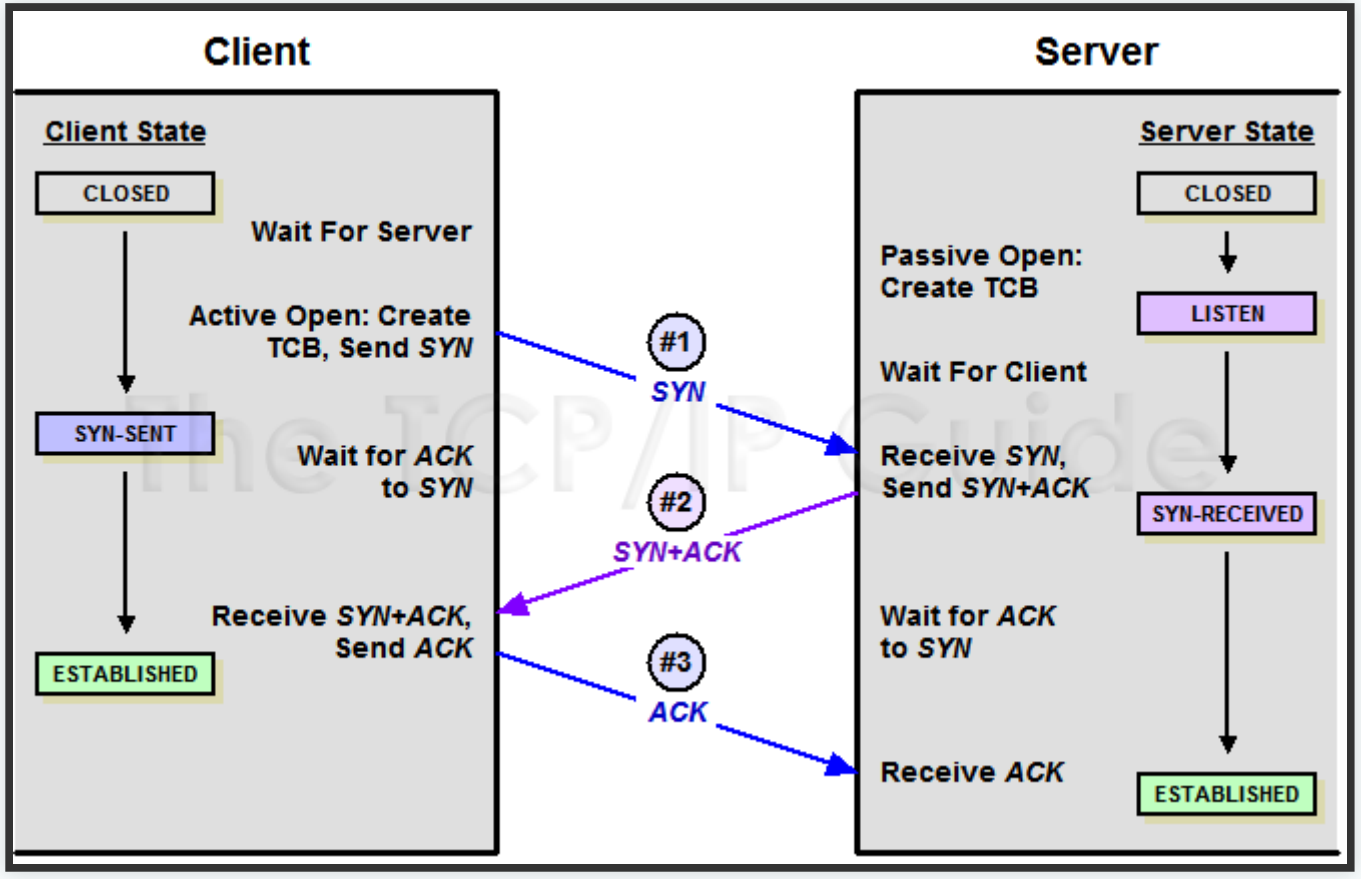
Ограничиваемые ресурсы

- core - limits the core file size (KB)
- data - max data size (KB)
- fsize - maximum filesize (KB)
- memlock - max locked-in-memory address space (KB)
- nofile - max number of open file descriptors
- rss - max resident set size (KB)
- stack - max stack size (KB)
- cpu - max CPU time (MIN)
- nproc - max number of processes

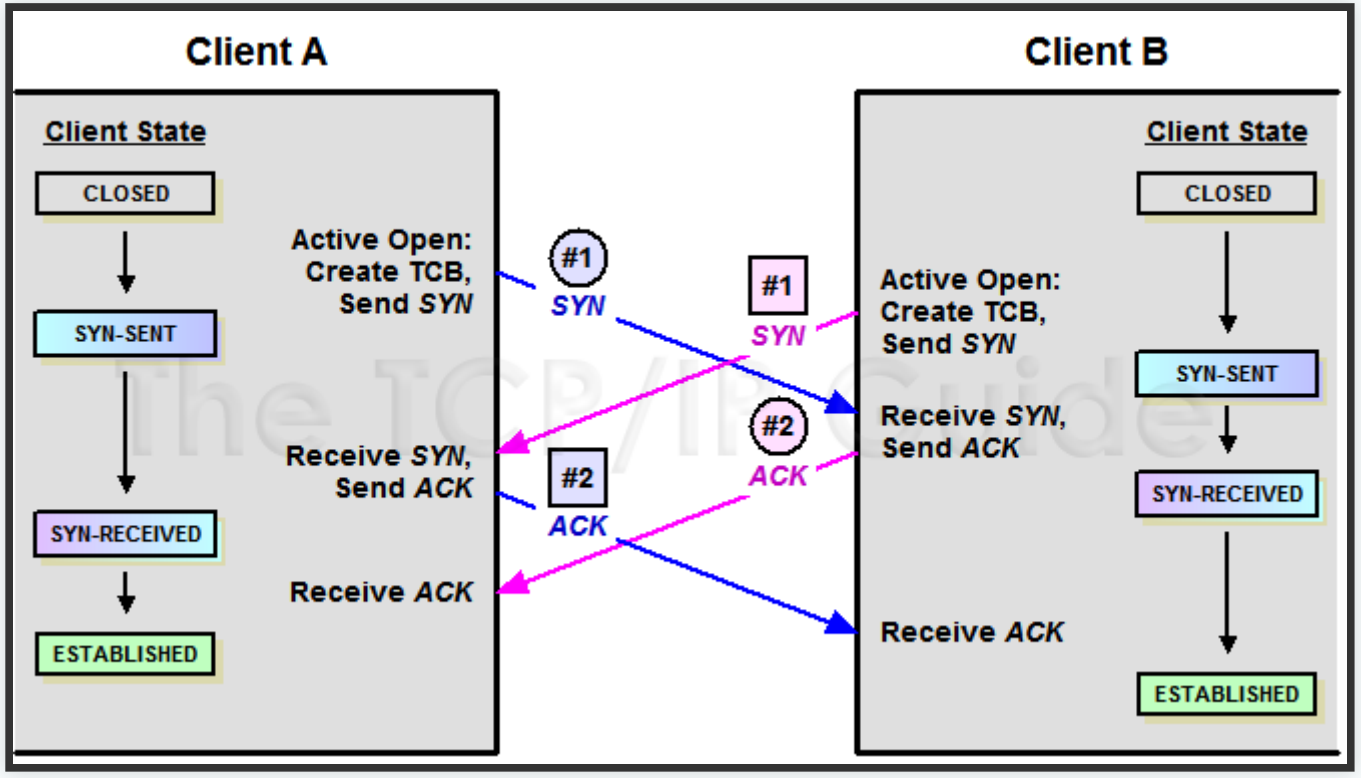
Ограничиваемые ресурсы

- as - address space limit (KB)
- maxlogins - max number of logins for this user
- maxsyslogins - max number of logins on the system
- priority - the priority to run user process with
- locks - max number of file locks the user can hold
- sigpending - max number of pending signals
- msgqueue - max memory used by POSIX message queues (bytes)
- nice - max nice priority allowed to raise to values: [-20, 19]
- rtprio - max realtime priority

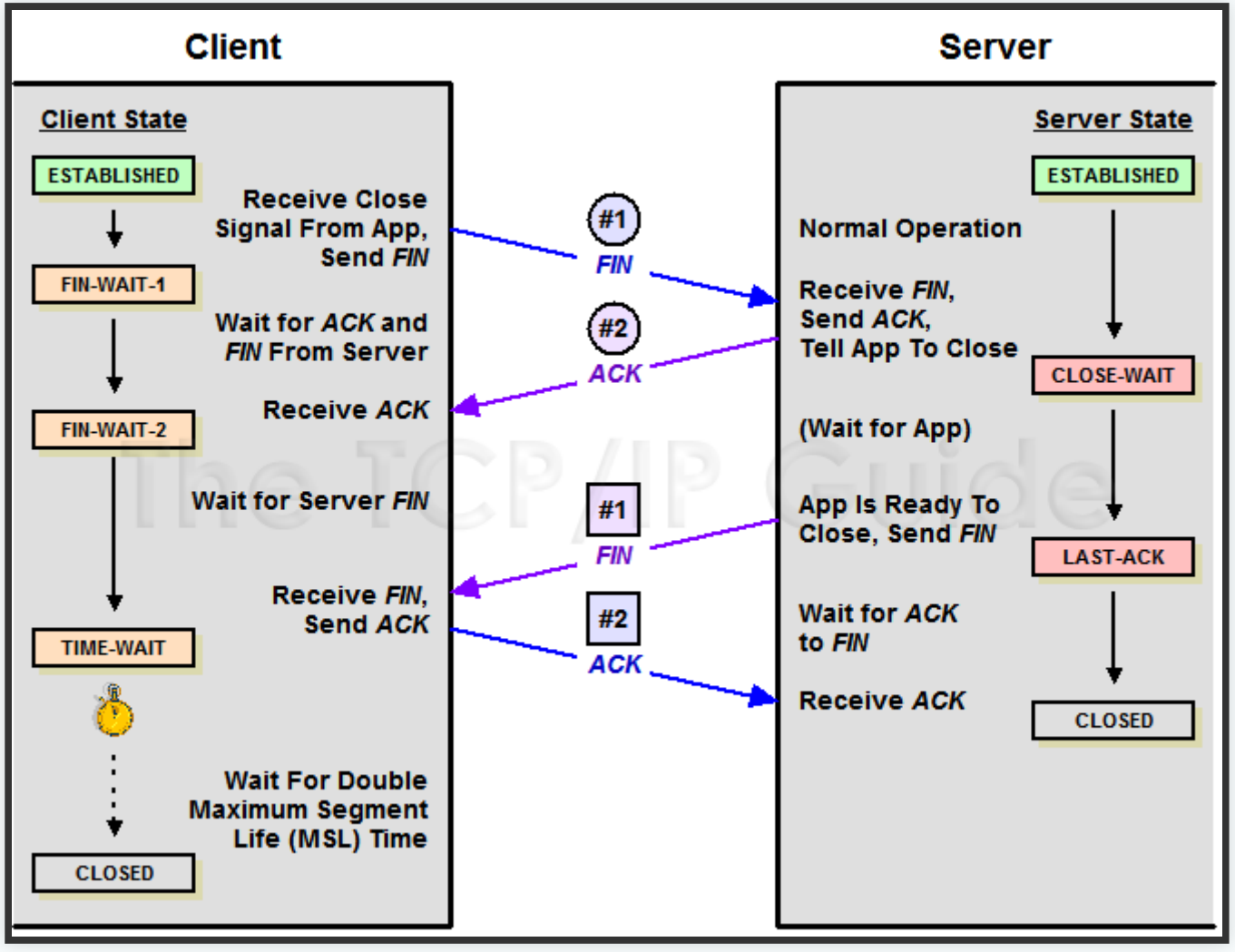
Тройное рукопожатие



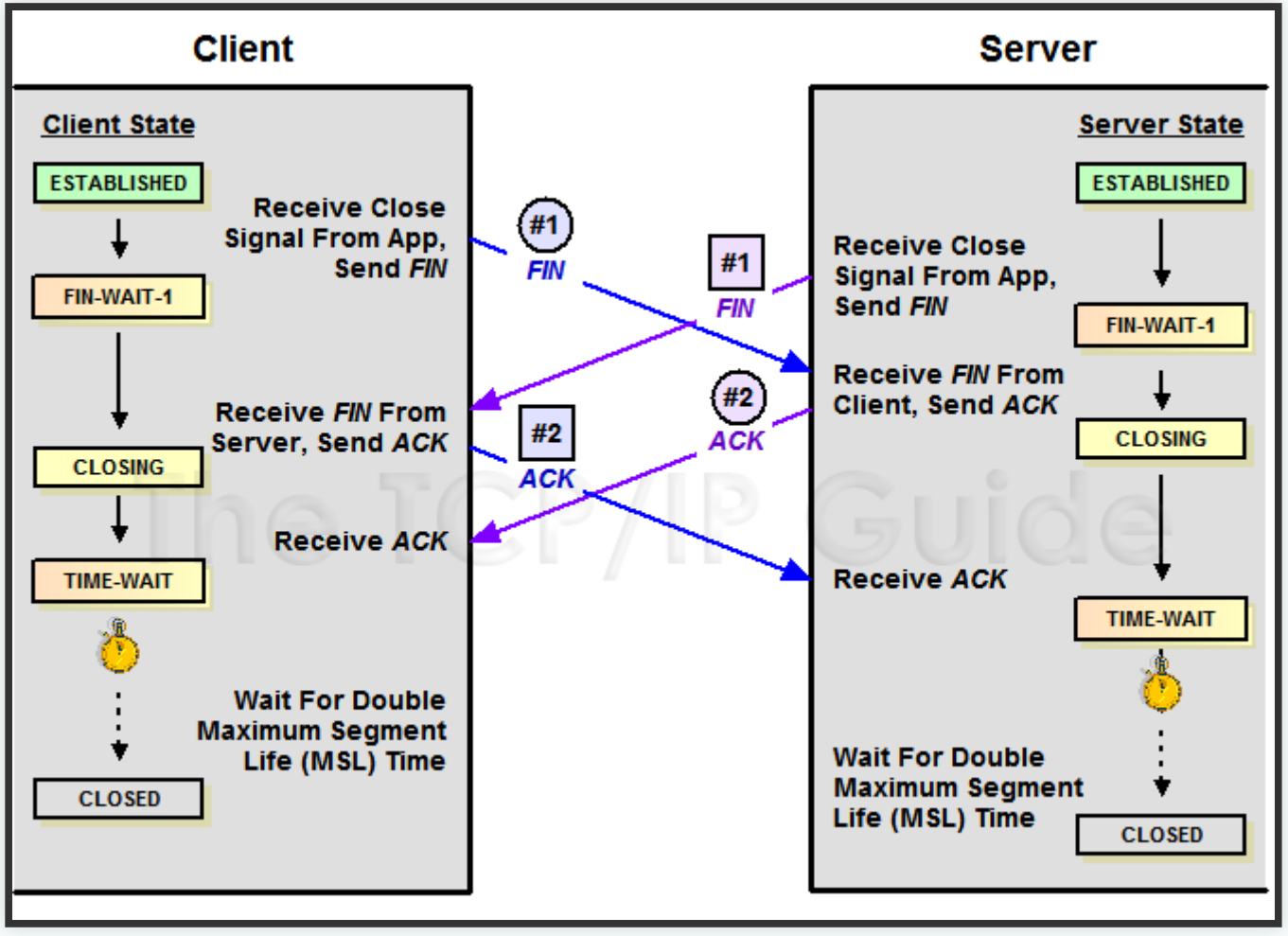
Одновременное открытие



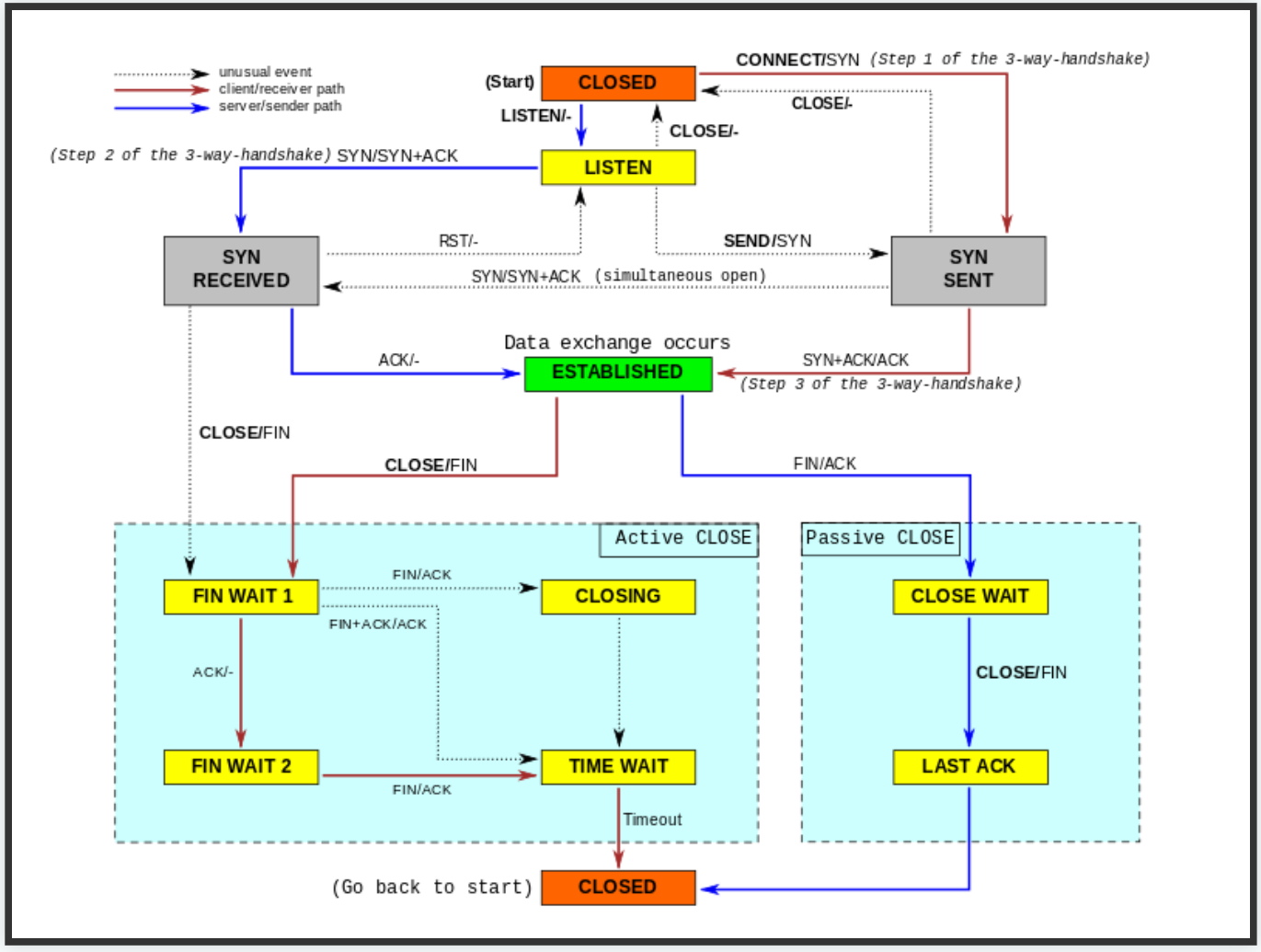
Закрытие соединения



Одновременное закрытие



Стейт машина протокола TCP



заголовок TCP

Бит	0 — 3	4 — 9	10 — 15	16 — 31
0	Порт источника, **Source Port**			Порт назначения, **Destination Port**
32	Порядковый номер, **Sequence Number (SN)**			
64	Номер подтверждения, **Acknowledgment Number (ACK SN)**			
96	Длина заголовка, (**Data offset**)	Зарезервировано	Флаги: CWR, ECE, URG, ACK, PSH, RST, SYN, FIN	Размер Окна, **Window size**
128	Контрольная сумма, **Checksum**			Указатель важности, **Urgent Point**
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

Общие термины

1. MTU - размер определяется стандартом (Ethernet II - 1500, WLAN 802.11 - 2272)
2. PMTU - устанавливается параметр DF, значение MTU определяется итеративно на основе ICMP пакетов об ошибках
3. MSS - MTU минус длинна заголовков TCP и IP
4. RTO - значение тайм-аута на отправку сегмента и получение ответа
5. PAWS - Детектирование повторного использования порядковых номеров удаляет сегменты со старыми метками времени
6. RTT - (Round Trip Time) интервал между отправкой покета и получением подтверждения обработки.

TCP Fast Open (TFO)

net.ipv4.tcp_fastopen

TCP Fast Open (TFO) – это механизм, который позволяет снизить задержку за счет того, что позволяет отправку данных внутри SYN-пакета. Однако и у него есть свои ограничения: в частности, на максимальный размер данных внутри SYN-пакета. Кроме того, только некоторые типы HTTP-запросов могут использовать TFO, и это работает только для повторных соединений, поскольку использует cookie-файл.

Window Size

Window Size определяет количество байт данных (payload), после передачи которых ожидается подтверждение от получателя.

Окно перегрузки

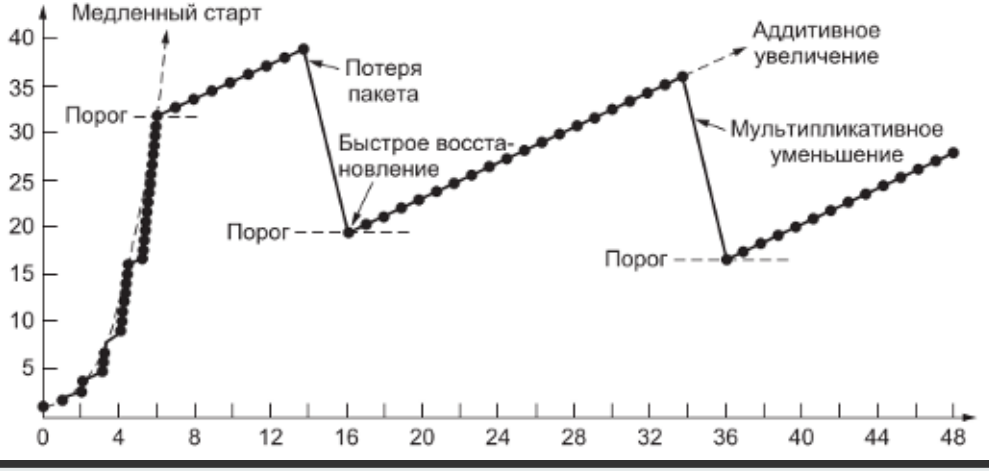
net.ipv4.tcp_window_scaling

Контроль перегрузки в TCP реализует AIMD с помощью окна, а в качестве сигнала используется потеря пакетов. В каждый момент времени в сети может находиться не более чем фиксированное число байт от каждого отправителя. Это число байт составляет размер окна перегрузки.

алгоритмы контроля перегрузкии

TCP Tahoe

TCP Reno



современные алгоритмы контроля перегрузкии

net.ipv4.tcp_congestion_control
net.ipv4.tcp_slow_start_after_idle

- BIC, CUBIC, Highspeed, H-TCP, NewReno, Illinois — эти алгоритмы созданы для так называемых long fat networks — длинных (а значит, с высоким RTT) и быстрых сетей.
- TCP Hybla здорово ведет себя на спутниковых линках
- Veno создан для беспроводных сетей с высокой потерей пакетов.

SACK

net.ipv4.tcp_sack

Выборочные подтверждения - это параметр передающийся в разделе optionsю в котором может соделжаться до 3 диапазонов успешно полученых байт.

Это позволяет точнее определять какие пакеты следует передать повторно

Orphans

net.ipv4.tcp_max_orphans
net.ipv4.tcp_orphan_retries

Когда у сокета вызывается метод close. это означает что у сокета уже нет файлового дискриптора и взаимодействие с сокетом не возможно, но он все равно висит в пямяти в качестве сироты.

Каждое orphan-соединение поглощает около 64Кб несбрасываемой на диск (unswappable) памяти.

Переиспользование сокетов в TIME-WAIT

net.ipv4.tcp_tw_reuse
net.ipv4.tcp_tw_recycle

- tcp_tw_reuse - разрешаем повторное использование TIME-WAIT сокетов в случаях, если протокол считает это безопасным
- tcp_tw_recycle - уменьшает время в TIME-WAIT с 2MSL до RTO. Эта опция сделает сервис недоступным для клиентов за NAT в случае если при трансляции «пропускаются» TCP-timestamp от клиентов

TimeStamps

net.ipv4.tcp_timestamps

Временные метки помогают проводить точное измерение времени обращения RTT для последующей корректировки значения таймера повторной передачи RTO.

В высоко скоростных сетях, где нмера быстро кончаются временные метки позволяют определять дубликаты

RFC 1337 TIME-WAIT Assassination Hazards in TCP

net.ipv4.tcp_rfc1337

- устаревший дубликат пакета с данными может быть ошибочно воспринят в новом соединении, что приведет к передаче неверных данных.
- соединение может быть десинхронизировано и уйти в ACK-цикл из-за устаревших дубликатов
- устаревшие дубликаты могут проникнуть в недавно созданное соединение и ошибочно уничтожить его

Очереди

net.ipv4.tcp_max_syn_backlog
net.core.somaxconn

- syn - очередь на установку соединенния
- accept - очередь уже установленных соединений

Если переполняется accept очередь то дропаются пакеты syn и ack если переполняется syn очередь то только syn запросы.

Syncookies

net.ipv4.tcp_syncookies

- Для того чтобы не терять запросы на установление соединения используются syncookie. Серверу придётся игнорировать все TCP опции (увеличенный размер окна, метки времени и др.), т.к. они отправляются в первоначальном SYN-запросе.
- SynCookie - техника противодействия SYN-флуд-атаке. В ответ на syn запрос сервер не ставит сокет в syn очередь до ожидания ack, а отправляет зашифрованый номер.

Mem

net.ipv4.tcp_mem
net.ipv4.tcp_rmem
net.ipv4.tcp_wmem

- tcp_mem - вся память выделяемая под протокол TCP измеряется в страницах памяти (минимум, режим нагрузки, максимум)
- tcp_rmem - приемный буфер, на основе значения по умолчанию и дополнительных параметров высчитывается размер окна
- tcp_wmem - буфер передачи

параметры защиты

- net.ipv4.conf.all.accept_redirects = 0
- net.ipv4.conf.all.secure_redirects = 0
- net.ipv4.conf.all.send_redirects = 0
- net.ipv4.tcp_max_orphans = 65536
- net.ipv4.tcp_orphan_retries = 0
- net.ipv4.conf.all.rp_filter = 1
- net.ipv4.conf.all.accept_source_route = 0
- net.ipv4.tcp_rfc1337 = 1
- net.ipv4.tcp_max_tw_buckets = 720000
- net.ipv4.ip_forward = 0
- net.ipv4.icmp_echo_ignore_broadcasts = 1
- net.ipv4.icmp_echo_ignore_all = 1

параметры оптимизации

- net.ipv4.tcp_fin_timeout = 10
- net.ipv4.tcp_keepalive_time = 1800
- net.ipv4.tcp_keepalive_intvl = 15
- net.ipv4.tcp_keepalive_probes = 5
- net.ipv4.tcp_max_syn_backlog = 4096
- net.ipv4.tcp_synack_retries = 1
- net.ipv4.netfilter.ip_conntrack_max = 16777216
- net.ipv4.tcp_timestamps = 1
- net.ipv4.tcp_sack = 1
- net.ipv4.tcp_fastopen = 1
- net.ipv4.tcp_slow_start_after_idle = 1
- net.ipv4.tcp_congestion_control = htcp

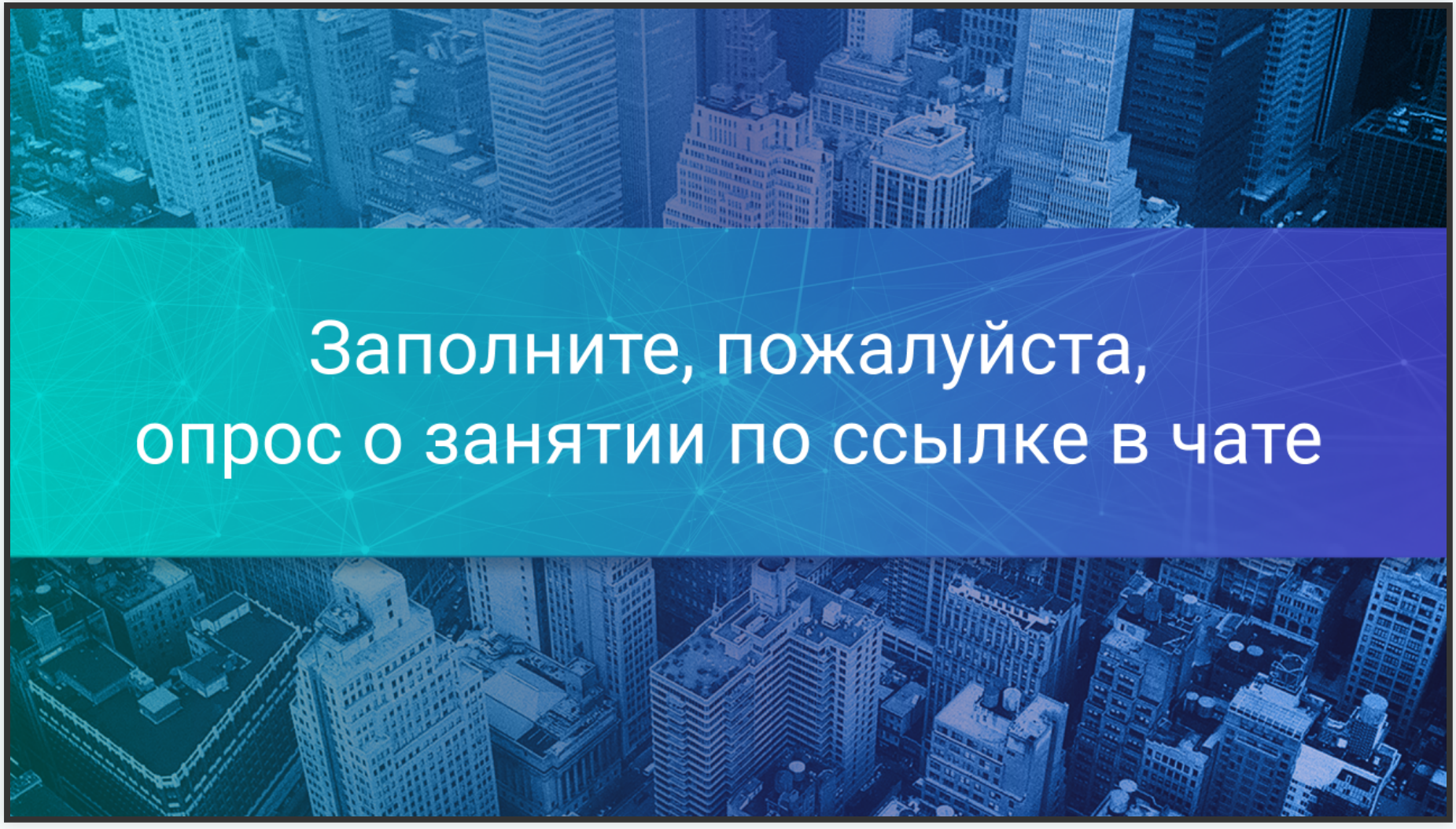
параметры оптимизации

- net.ipv4.tcp_no_metrics_save = 1
- net.ipv4.ip_local_port_range = 1024 65535
- net.ipv4.tcp_window_scaling = 1
- net.core.somaxconn = 65535
- net.core.netdev_max_backlog = 1000
- fs.file-max = 64000
- net.ipv4.tcp_mem = 50576 64768 98152
- net.ipv4.tcp_rmem = 4096 87380 16777216
- net.ipv4.tcp_wmem = 4096 65536 16777216

Рефлексия

Отметьте 3 пункта, которые вам запомнились с вебинара

Что вы будете применять в работе из сегодняшнего вебинара?


Заполните, пожалуйста,
опрос о занятии по ссылке в чате