Лекция 3. Сетевое программирование. Ликбез по проектированию HighLoad систем

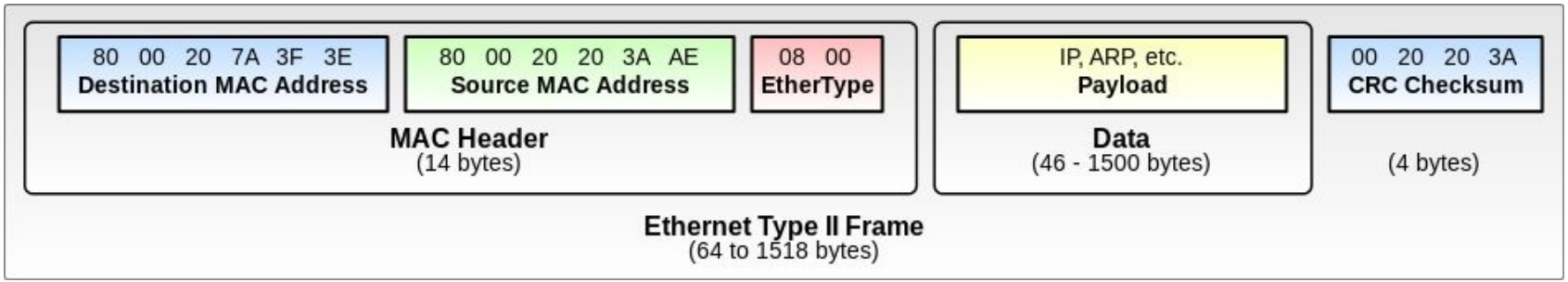
- О преподавателе и курсе.
- Понятие высокой нагрузки.
- Трехзвенная архитектура.
- Типы конкурентной обработки клиентов.
- Масштабирование.
- Кеширование.
- Очереди.

Модель OSI

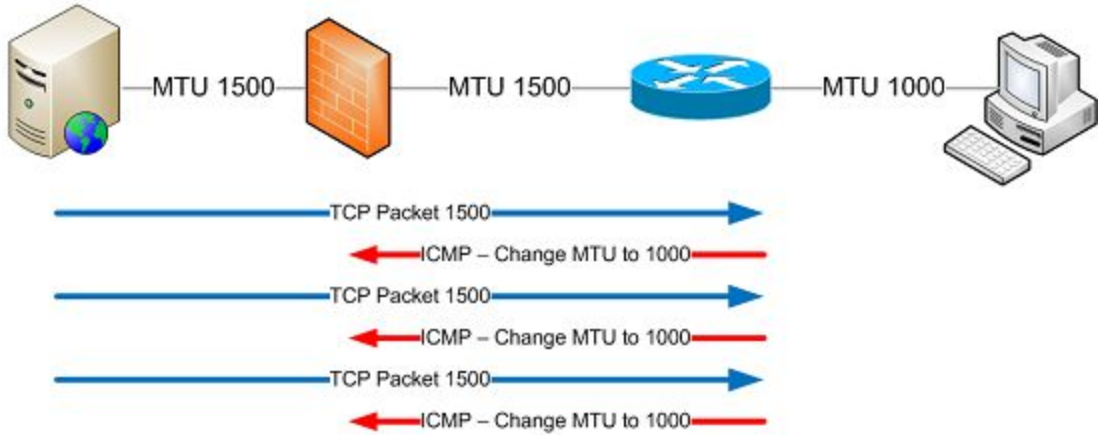

Модель OSI
Данные
Прикладной
доступ к сетевым службам
Данные
Представления
представление и кодирование данных
Данные
Сеансовый
Управление сеансом связи
Блоки
Транспортный
безопасное и надёжное соединие точка-точка
Пакеты
Сетевой
Определение пути и IP (логическая адресация)
Кадры
Канальный
MAC и LLC
(Физическая адресация)
Биты
Физический
кабель, сигналы, бинарная передача данных

Модель TCP/IP

Ethernet


80 00 20 7A 3F 3E
Destination MAC Address
80 00 20 20 3A AE
Source MAC Address
08 00
EtherType
MAC Header
(14 bytes)
IP, ARP, etc.
Payload
Data
(46 - 1500 bytes)
00 20 20 3A
CRC Checksum
(4 bytes)
Ethernet Type II Frame
(64 to 1518 bytes)

Path MTU Discovery

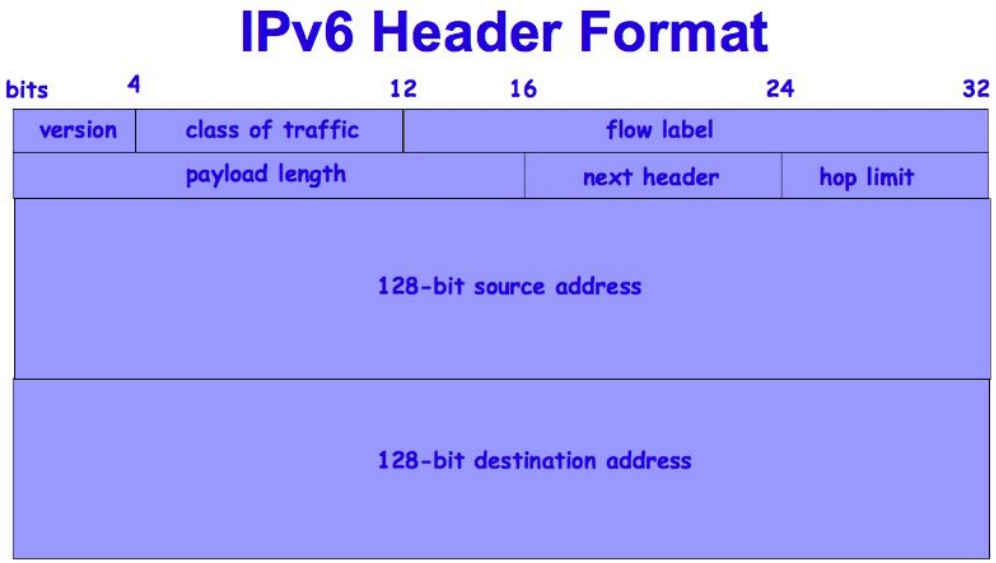


IPv4

IPv4 Header Format

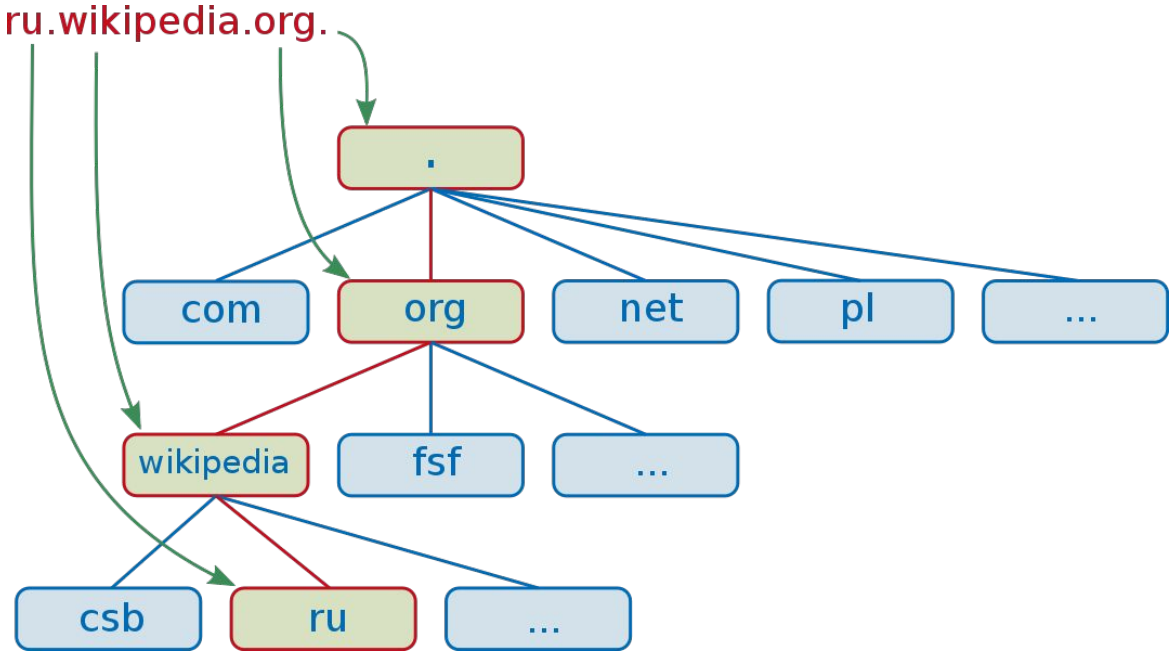
Отступ	Октет	0								1								2								3							
Октет	Бит	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
0	0	Версия				Размер заголовка				Differentiated Services Code Point						Explicit Congestion Notification		Размер пакета (полный)															
4	32	Идентификатор																Флаги			Смещение фрагмента											
8	64	Время жизни								Протокол								Контрольная сумма заголовка															
12	96	IP-адрес источника																															
16	128	IP-адрес назначения																															
20	160	Опции (если размер заголовка > 5)																															
20 или 24+	160 или 192+	Данные																															

IPv6

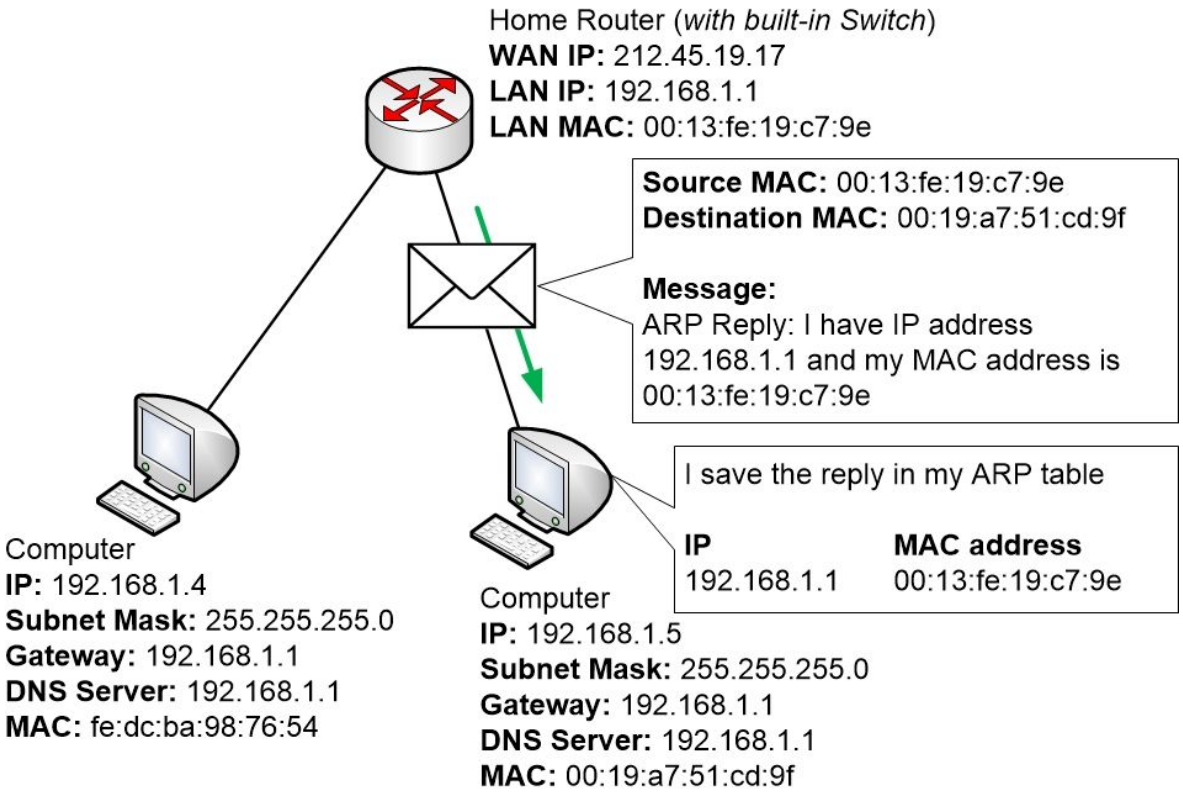


Total length: 40 bytes

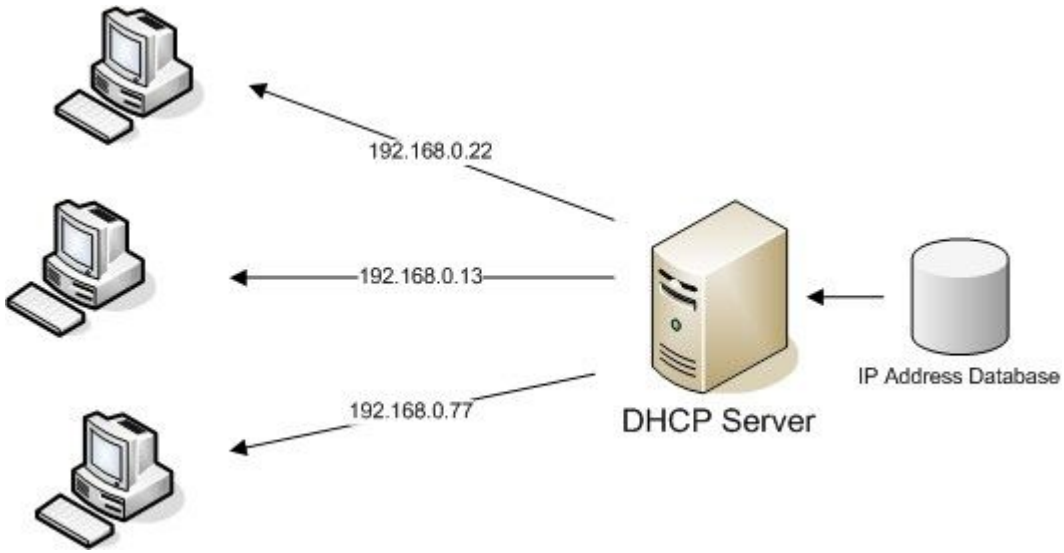
DNS



ARP


Home Router (*with built-in Switch*)
WAN IP: 212.45.19.17
LAN IP: 192.168.1.1
LAN MAC: 00:13:fe:19:c7:9e
Source MAC: 00:13:fe:19:c7:9e
Destination MAC: 00:19:a7:51:cd:9f
Message:
ARP Reply: I have IP address 192.168.1.1 and my MAC address is 00:13:fe:19:c7:9e
I save the reply in my ARP table
IP 192.168.1.1
MAC address 00:13:fe:19:c7:9e
Computer
IP: 192.168.1.4
Subnet Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS Server: 192.168.1.1
MAC: fe:dc:ba:98:76:54
Computer
IP: 192.168.1.5
Subnet Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS Server: 192.168.1.1
MAC: 00:19:a7:51:cd:9f

DHCP


192.168.0.22
192.168.0.13
192.168.0.77
DHCP Server
IP Address Database

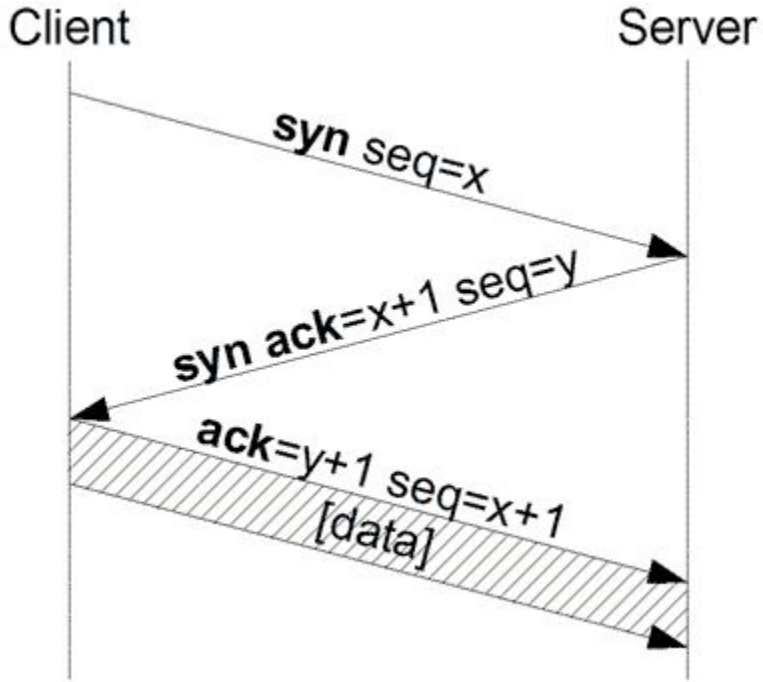
UDP

Биты	0 - 15	16 - 31
0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
64-...	Данные (Data)	

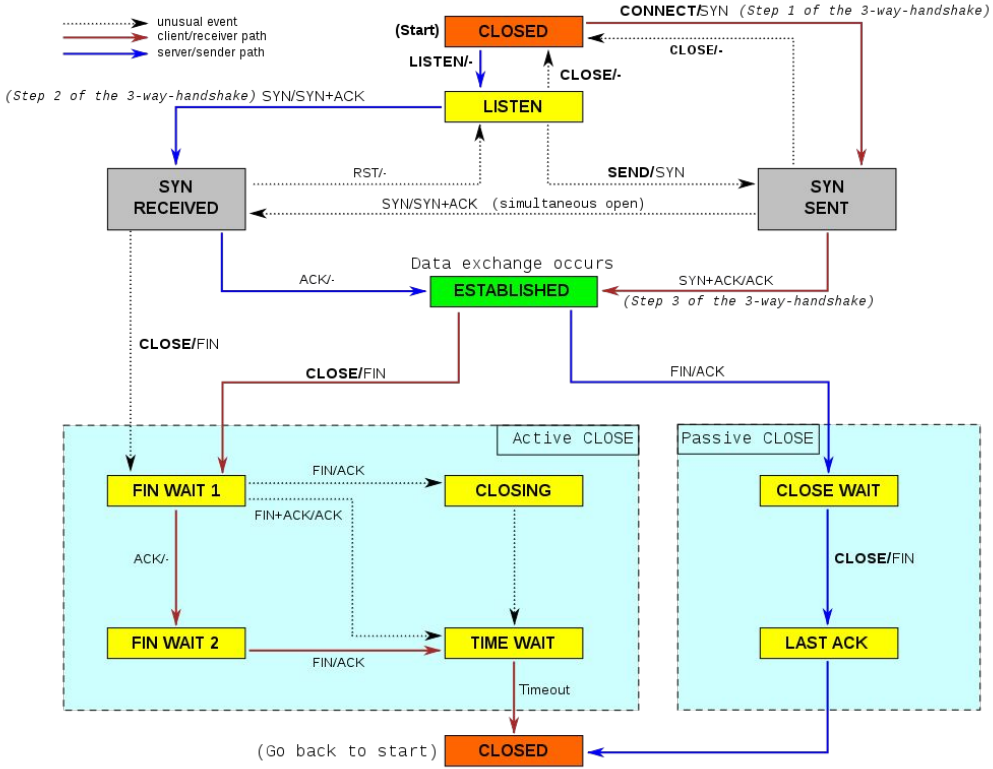
TCP

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	Зарезервировано	Флаги	Размер Окна
128	Контрольная сумма			Указатель важности
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

TCP: установка соединения



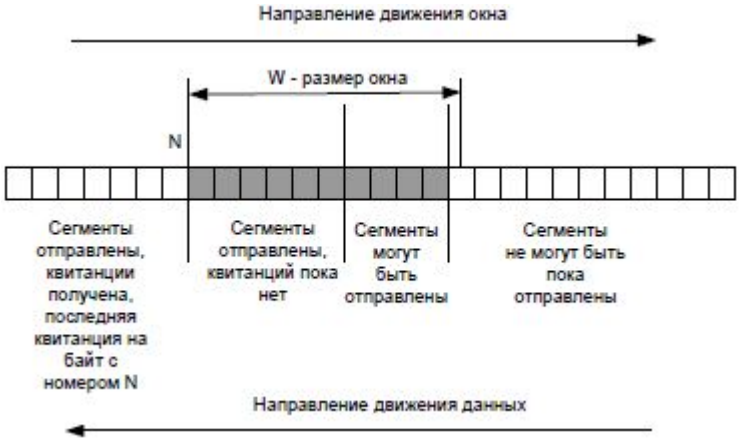
TCP: состояния



TCP: состояния

Состояния сеанса TCP	
CLOSED	Начальное состояние узла. Фактически фиктивное
LISTEN	Сервер ожидает запросов установления соединения от клиента
SYN-SENT	Клиент отправил запрос серверу на установление соединения и ожидает ответа
SYN-RECEIVED	Сервер получил запрос на соединение, отправил ответный запрос и ожидает подтверждения
ESTABLISHED	Соединение установлено, идёт передача данных
FIN-WAIT-1	Одна из сторон (назовём её узел-1) завершает соединение, отправив сегмент с флагом FIN
CLOSE-WAIT	Другая сторона (узел-2) переходит в это состояние, отправив, в свою очередь сегмент ACK и продолжает одностороннюю передачу
FIN-WAIT-2	Узел-1 получает ACK, продолжает чтение и ждёт получения сегмента с флагом FIN
LAST-ACK	Узел-2 заканчивает передачу и отправляет сегмент с флагом FIN
TIME-WAIT	Узел-1 получил сегмент с флагом FIN, отправил сегмент с флагом ACK и ждёт 2*MSL секунд, перед окончательным закрытием соединения
CLOSING	Обе стороны инициировали закрытие соединения одновременно: после отправки сегмента с флагом FIN узел-1 также получает сегмент FIN, отправляет ACK и находится в ожидании сегмента ACK (подтверждения на свой запрос о разъединении)

Окно TCP


Направление движения окна
W - размер окна
N
Сегменты отправлены, квитанции получена, последняя квитанция на байт с номером N
Сегменты отправлены, квитанций пока нет
Сегменты могут быть отправлены
Сегменты не могут быть пока отправлены
Направление движения данных

Алгоритм Нейгла

```
if there is new data to send
  if the window size >= MSS and available data is >= MSS
    send complete MSS segment now
  else
    if there is unconfirmed data still in the pipe
      enqueue data in the buffer until an acknowledge is received
    else
      send data immediately
    end if
  end if
end if
```

Маска подсети

IP-адрес:	11000000 10101000 00000001 00000010	(192.168.1.2)
Маска подсети:	11111111 11111111 11111110 00000000	(255.255.254.0)
Адрес сети:	11000000 10101000 00000000 00000000	(192.168.0.0)

HTTP

```
1  HTTP/1.1 200 OK
2  Date: Mon, 24 Jul 2017 04:11:15 GMT
3  Server: Apache/2.4.7 (Ubuntu)
4  X-Powered-By: PHP/5.5.9-1ubuntu4.21
5  Vary: Accept-Encoding
6  Content-Length: 2548
7  Content-Type: text/html
```

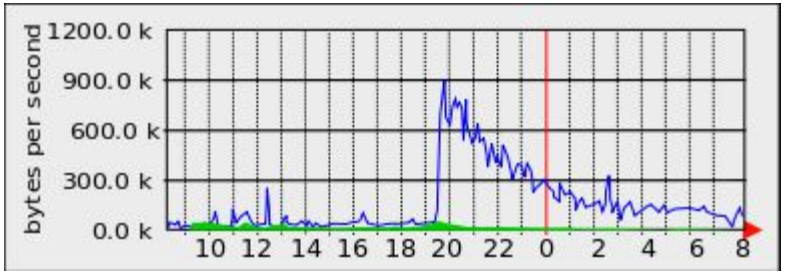
HTTP 2.0

- Бинарный протокол.
- Кеширование заголовков.
- Сжатие заголовков.
- Мультиплексирование.

Факторы нагрузки

- rps (rpm)
- Число одновременных соединений (C10k, C100k)
- bytes per second

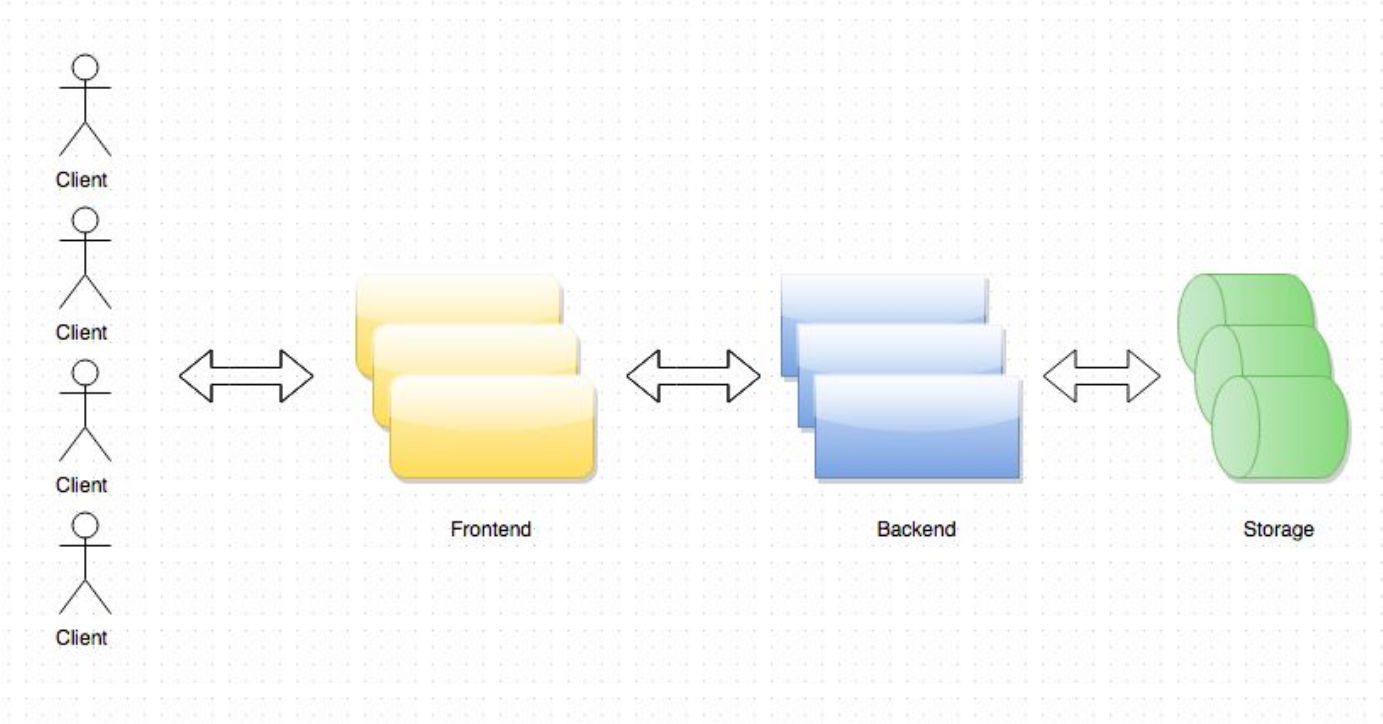
Slashdot effect



Latency and throughput

Latency - время обслуживания одного клиента
Throughput - число обработанных клиентов в единицу времени

Трехзвенная архитектура



Frontend

- SSL терминирование
- Обработка медленных клиентов
- Отдача статики.
- Keep-Alive
- Кэширование (*)

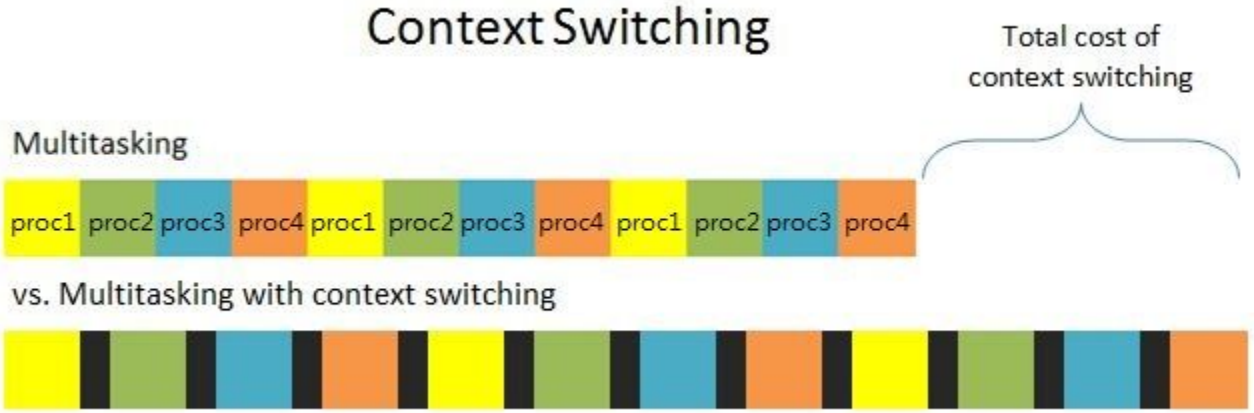
Backend

- Бизнес-логика (как правило, для веба она очень простая).
- Ожидание ответов от баз данных.

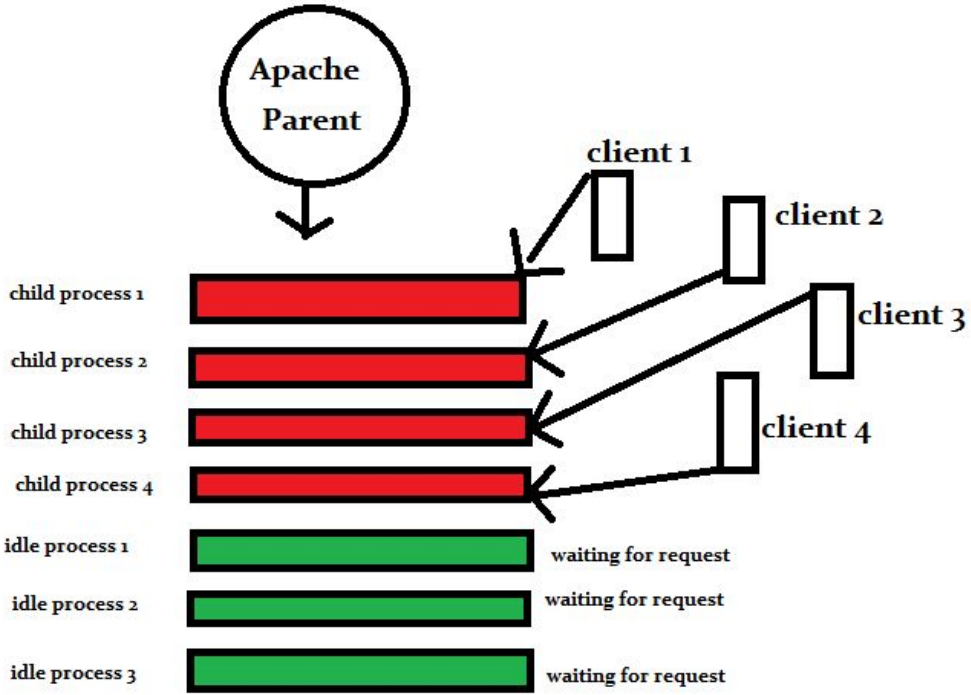
Числа, которые должен знать каждый программист

Обращение к кэшу L1	0.5 нс
Ошибка при предсказании условного перехода	5 нс
Обращение к кэшу L2	7 нс
Открытие/закрытие мьютекса	25 нс
Обращение к главной памяти	100 нс
Сжатие 1 Кб быстрым алгоритмом	3,000 нс
Пересылка 2Кб по сети со скоростью 1 Гб/с	20,000 нс
Чтение 1 Мб последовательно из главной памяти	250,000 нс
Передача сообщения туда/обратно в одном дата-центре	500,000 нс
Произвольный доступ к жёсткому диску	10,000,000 нс
Чтение 1 Мб последовательно с жёсткого диска	20,000,000 нс
Передача пакета из Калифорнии в Нидерланды и обратно	150,000,000 нс

Context Switch



Многопроцессная/многопоточная обработка



Асинхронная обработка

- epoll/kqueue
- Обработка в 1 процесс
- Context-Switch отсутствует.
- event-driven управление, основанное на callback.

Вопрос

Почему статику можно эффективно отдавать фронтендом, несмотря на однопоточность?

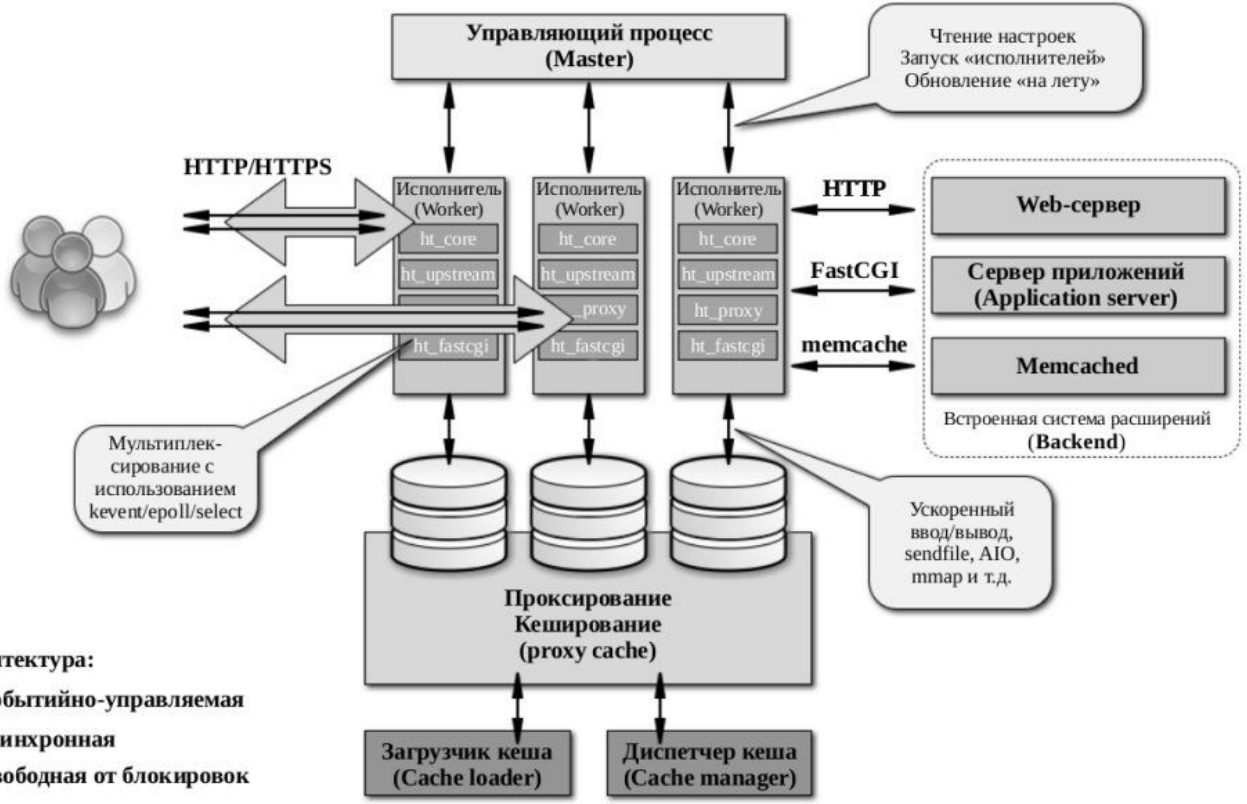
Вопрос

Какие проблемы есть у однопроцессного (однопоточного) демона?

Кооперативная многозадачность

- “Потоки” сами решают, когда передавать управление.
- По сути, это почти асинхронное поведение, но программируется также просто, как синхронное.

Архитектура nginx



Архитектура:

- Событийно-управляемая
- Асинхронная
- Свободная от блокировок

DNS-балансировка

MBP-Remen:~ bhychik$ host mail.ru
mail.ru has address 94.100.180.201
mail.ru has address 94.100.180.199
mail.ru has address 217.69.139.199
mail.ru has address 217.69.139.200
mail.ru has IPv6 address 2a00:1148:db00:0:b0b0::1
mail.ru mail is handled by 10 mxs.mail.ru.

MBP-Remen:~ bhychik$ host mail.ru
mail.ru has address 217.69.139.200
mail.ru has address 94.100.180.201
mail.ru has address 94.100.180.199
mail.ru has address 217.69.139.199
mail.ru has IPv6 address 2a00:1148:db00:0:b0b0::1
mail.ru mail is handled by 10 mxs.mail.ru.
MBP-Remen:~ bhychik$

Горизонтальное масштабирование

- Добавление новых серверов.
- На определенной стадии дешевле вертикального масштабирования.
- Сложнее программировать.

Вертикальное масштабирование

- Более мощное железо в существующие сервера
- Не надо программировать.
- Нелинейный рост цены.
- Все равно ограничены сверху мощностью топового железа на рынке.

Кеширование

- На клиенте.
- На фронтендах.
- На выделенных серверах (memcached/redis/tarantool)

Инвалидация кеша

- Тегирование.
- Двойное чтение.
- Старт с непрогретым кешом
- Cache misses.

Очереди

- RabbitMQ/ActiveMQ
- Удобно, когда действие нужно не мгновенно.