РОССИЙСКИЕ ИНТЕРНЕТ-ТЕХНОЛОГИИ
ВЫСОКИЕ НАГРУЗКИ

Что такое нагрузка?

Анатолий Орлов

anatolix@yandex-team.ru

Яндекс

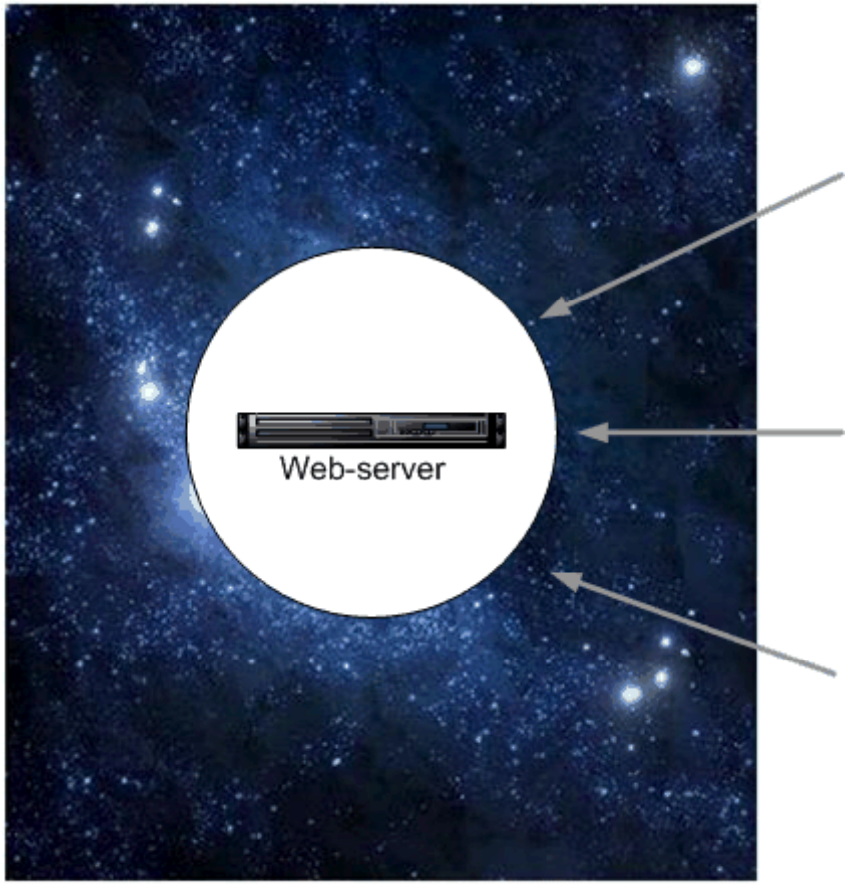
Disclaimer

- Доклад “попсовый”- содержит то, что все и так должны бы знать.
- Будет информация о железе, но без конкретики.

Цель:

- описать поведение базовых компонентов системы под нагрузкой.
- дать понимание, почему почти все сервера устроены так, а не иначе.

Сферический web-сервер в вакууме



- 50 запросов в секунду, время ответа 30ms
- 500 запросов в секунду, время ответа ??? ms
- 5000 запросов в секунду, ?????

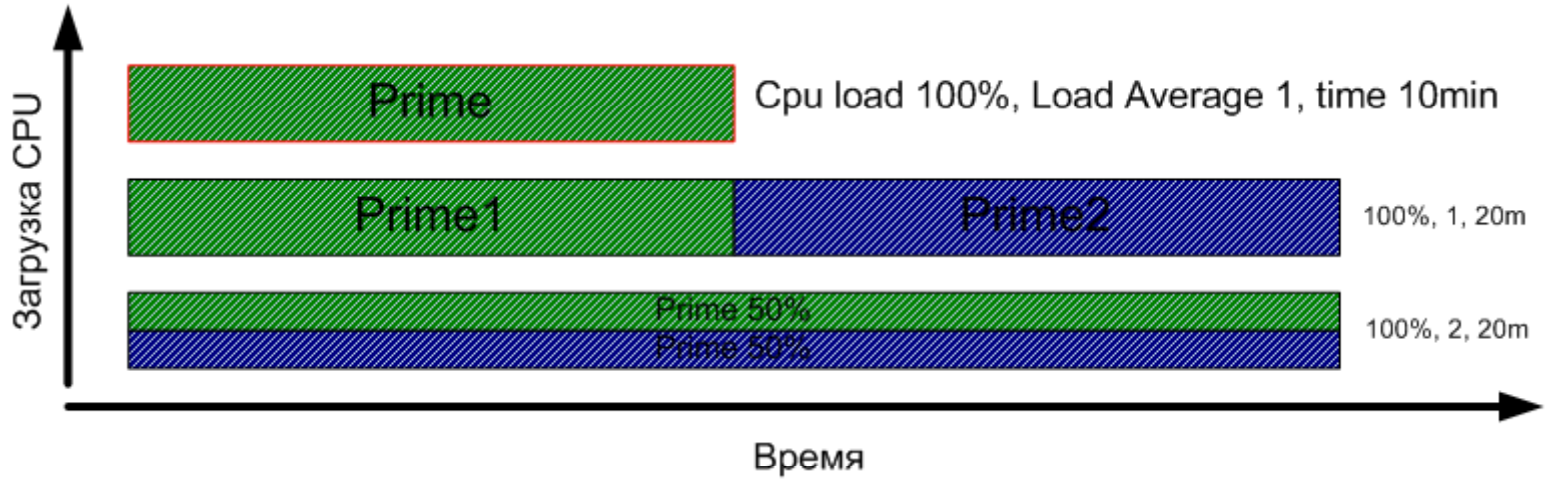
CPU – степень загрузки

Выдача команды “top” (FreeBSD)

```
last pid: 74625;  load averages:  1.78,  1.68,  1.65   up 100+00:35:09 17:44:20
139 processes: 1 running, 137 sleeping, 1 zombie
CPU states: 29.0% user,  0.0% nice,  0.6% system,  0.1% interrupt, 70.4% idle
Mem: 15G Active, 126M Inact, 399M Wired, 53M Cache, 215M Buf, 7728K Free
Swap: 1024M Total, 505M Used, 519M Free, 49% Inuse
  PID USERNAME      THR PRI NICE   SIZE    RES STATE  C   TIME   WCPU COMMAND
53771 websearch      82  44    0 25339M 15173M ucond  3  88.7H 165.92% httpsear
19577 bind           11  44    0 41344K 15252K select 2   5:46  0.10% named
28878 monitor         3  44    0 35144K  3776K select 4  25:57  0.00% python2.5
  845 root            1  44    0  9404K   792K select 1   4:18  0.00% ntpd
  874 root            1  44    0 21740K   824K select 3   3:46  0.00% sshd
  880 root            1  44    0 10660K   912K select 0   1:59  0.00% sendmail
  686 root            1  44    0  5648K   576K select 3   0:52  0.00% syslogd
37423 root            1   8    0  6708K   448K nanslp 2   0:26  0.00% cron
```

Load average

Load Average – среднее число программ, претендующих на выполнение.



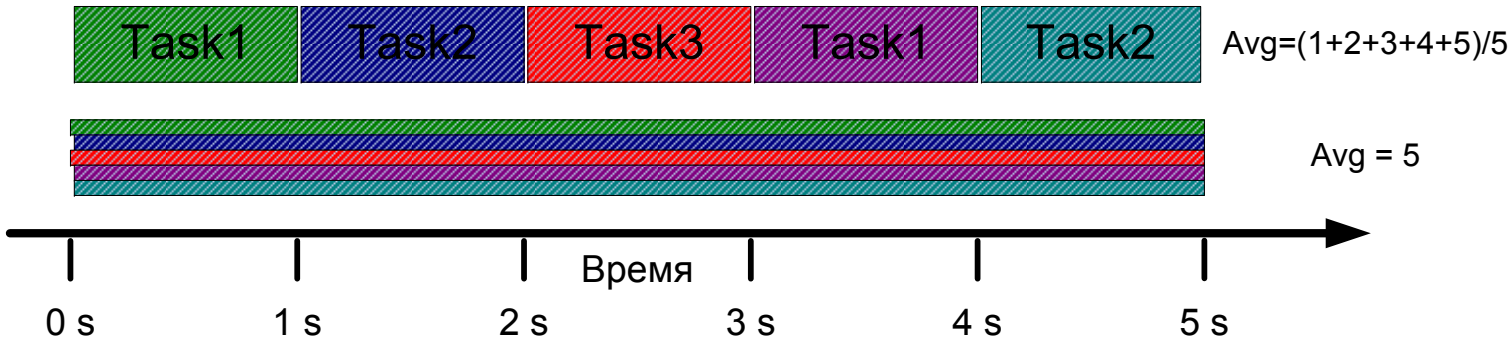
Вывод: Load Average непосредственно влияет на скорость выполнения конкретной задачи.

Очереди vs. Параллельное выполнение

- **Очереди выполнения – простой и дешевый способ уменьшения Load Average(на общее время выполнения не влияет).**

Зачем уменьшать LA:

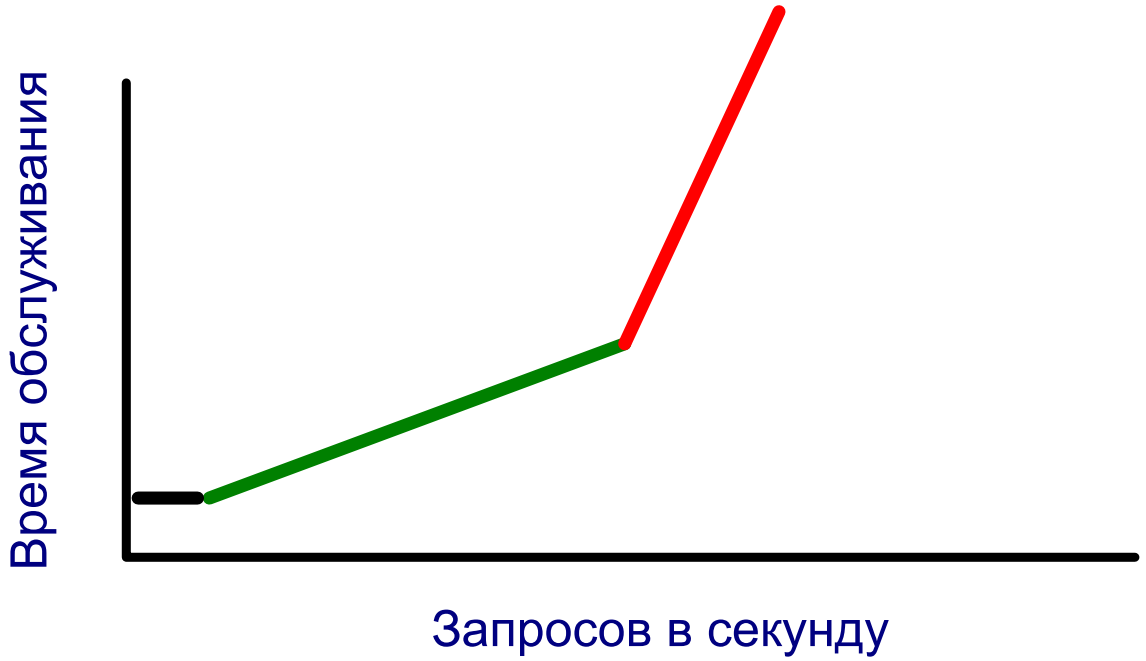
- **[спорное] мнение, что операционные системы плохо переносят большое количество потоков.**
- **Положительное влияние на среднее время ответа***



- **Вспоминая про SMP: параллельное выполнение все равно нужно.**

Возвращаясь к web серверам

- Скорость ответа зависит от количества запросов в секунду.
- Много запросов -> либо большой load average, либо большая очередь запросов.



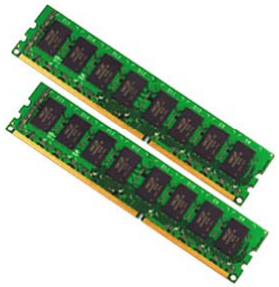
“Реалтайм” vs. “числомолотилки”

Какая должна быть нагрузка на CPU?

- Задачи “типа реалтайм” – цель минимизация среднего времени ответа.
- Задачи “числомолотилка”- цель максимизация количества обработанных данных в единицу времени.

Наблюдение: сочетание задач обоих типов на одной машине счастья не приносит.

Оперативная память



Про память рассказывать почти нечего:

- Скорость работы памяти прогнозируема, плохо поддается влиянию программиста, часто считается постоянной.
- Работа с памятью считается частью работы CPU -> Нет отдельных системных программ типа top.

Нюансы:

- Не является полностью однородной из-за наличия кэшей. Профайлеры умеют профайлить работу с кэшами. cache-aware алгоритмы.
- Есть аспекты, связанные с одновременным доступом к одной и той же страничке из разных CPU.

Диск

Насколько диск медленней памяти?

- Приблизительно в 17 раз
- Приблизительно в 14000 раз

Диск vs Память

Диск (SATA2-300)

- Average Seek Time – 5ms
- Peak Transfer – 375Mb/s
- Механика диска:
  - inner 44-72 Mb/s
  - outer 74-111Mb/s

Память(DDR3-800)

- 7-7-7-15 * 10ns
- Peak Transfer 6400

Скорость диска на практике

- Кэширование OS и самого диска
- Read Ahead и IO Scheduling
- Влияние RAID на скорость
- SAS vs. SATA (SCSI vs. IDE)

SSD – ни рыба, ни мясо

Сама по себе флеш память:

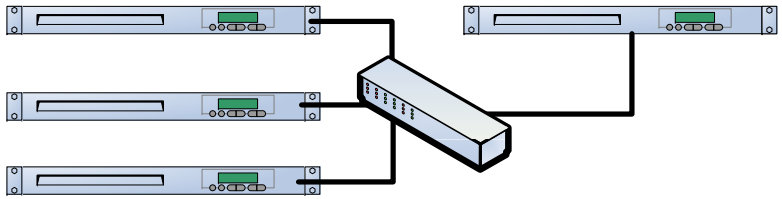
- **Является random access – т.е. нету seek;**
- **Быстро читает, но медленно пишет;**
- **Имеет ограниченное количество циклов записи;**
- **Дорогая.**

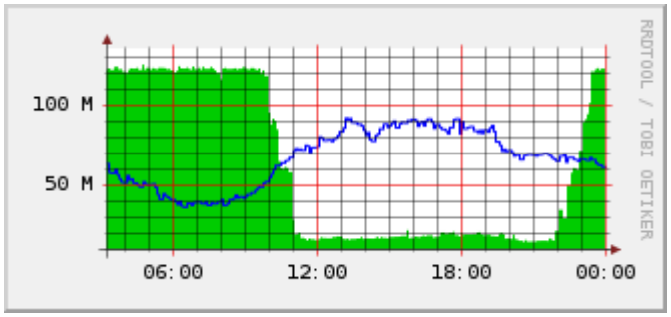
Есть типа SSD девайсов:

- **“Типа как HDD, только быстрее” (интерфейс SATA/SAS, форм фактор как у диска). Быстрый seek, читает сравнимо, пишет медленно.**
- **“Типа как память только persistent” (интерфейс PCI-E).**

Сеть

- Данные передаются пакетами, иногда нагрузку лучше измерять в пакетах, а не в мегабайтах.
- Сеть асинхронная (100mb/s = 100 туда и 100 обратно).
- Скорость сети, как правило, зависит от центральных узлов (роутеры, коммутаторы). Congestion points. Очереди.





Говорим “сеть” – думаем “та сторона”

Большая часть ожидания сети - это ожидание программы с другой стороны.

- Ожидание броузера на медленном канале;
- Ожидание mysql, который выполняет запрос;
- Etc.

Как правило, сделать с этим с нашей стороны ничего нельзя.

Но можно минимизировать ресурсы захваченные с нашей стороны на время ожидания.

- Обработка многих connection-ов в одном потоке.

Итого: что должно быть в сервере

- **Должен работать в несколько потоков/процессов.**
  чтобы использовать возможности SMP
- **Должен содержать очереди.**
  чтобы снижать LA и чуть-чуть улучшать время ответа.
  note: очереди у вас все равно есть, хотите вы или нет.
- **Желательно уметь в одном потоке обрабатывать много сетевых соединений.**
  чтобы не тратить слишком много ресурсов из-за тормозов других.

Архитектура сферического веб-сервера в вакууме

- Один или несколько потоков, которые принимают соединения и кладут в очередь(select/epoll/kqueue).
- Пул потоков worker-ов, которые берут задачи из очереди, выполняют и кладут в очередь на отдачу. Формула Димы Тейблюма - #CPU+#Disk+5.
- Один или несколько потоков, которые отдают ответы по сети.

В такой архитектуре:

- **50rps – нормальная работа.**
- **500rps – очереди длинные – вырастает время ответа**
- **5000rps – worker-ы не смогут разгрести очередь, поэтому часть соединений отшибается.**