

HighLoad для начинающих

Dmitry E. Oboukhov

2014-10-07

HighLoad, что это?

- ▶ Конференция?

HighLoad, что это?

- ▶ Конференция?
- ▶ Высокая нагрузка?

HighLoad, что это?

- ▶ Конференция?
- ▶ Высокая нагрузка?
- ▶ Миф?!

Высокая нагрузка что это?

Высокая нагрузка что это?

- ▶ $53.328 * 10^9$ запросов в секунду?

Высокая нагрузка что это?

- ▶ $53.328 * 10^9$ запросов в секунду? - средний CPU

Высокая нагрузка что это?

- ▶ $53.328 * 10^9$ запросов в секунду? - средний CPU
- ▶ Более реалистично!

Высокая нагрузка что это?

- ▶ $53.328 * 10^9$ запросов в секунду? - средний CPU
- ▶ Более реалистично!
- ▶ 1 запрос в секунду?

Высокая нагрузка что это?

- ▶ $53.328 * 10^9$ запросов в секунду? - средний CPU
- ▶ Более реалистично!
- ▶ 1 запрос в секунду? - любой веб сервер?...

Высокая нагрузка что это?

- ▶ $53.328 * 10^9$ запросов в секунду? - средний CPU
- ▶ Более реалистично!
- ▶ 1 запрос в секунду? - любой веб сервер?...
перекодирующий видеоролики? :)

Высокая нагрузка это:

Высокая нагрузка это:

Нагрузка, с которой не справляется
железо

Когда это бывает?

Когда это бывает?

Достигнуты технические ограничения

Когда это бывает?

Достигнуты технические ограничения

- ▶ Сеть

Когда это бывает?

Достигнуты технические ограничения

- ▶ Сеть - за рамками данного доклада

Когда это бывает?

Достигнуты технические ограничения

- ▶ Сеть - за рамками данного доклада
- ▶ Память

Когда это бывает?

Достигнуты технические ограничения

- ▶ Сеть - за рамками данного доклада
- ▶ Память
- ▶ CPU

Причины

Причины

- ▶ Недоиспользование железа

Причины

- ▶ Недоиспользование железа
- ▶ Трудности масштабирования

Причины

- ▶ Недоиспользование железа
- ▶ Трудности масштабирования

Архитектурные проблемы

Недоиспользование железа

Рассмотрим типичный вебсервер.

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.
- ▶ Парсер запроса http.

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.
- ▶ Парсер запроса http.
- ▶ Валидация запроса, выбор контроллера.

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.
- ▶ Парсер запроса http.
- ▶ Валидация запроса, выбор контроллера.
- ▶ Запросы к хранилищу данных.

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.
- ▶ Парсер запроса http.
- ▶ Валидация запроса, выбор контроллера.
- ▶ Запросы к хранилищу данных.
- ▶ Формирование ответа (template)

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.
- ▶ Парсер запроса http.
- ▶ Валидация запроса, выбор контроллера.
- ▶ Запросы к хранилищу данных.
- ▶ Формирование ответа (template)
- ▶ Отправка ответа клиенту.

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.
- ▶ Парсер запроса http.
- ▶ Валидация запроса, выбор контроллера.
- ▶ Запросы к хранилищу данных.
- ▶ Формирование ответа (template)
- ▶ Отправка ответа клиенту.

Традиционная реализация

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.
- ▶ Парсер запроса http.
- ▶ Валидация запроса, выбор контроллера.
- ▶ Запросы к хранилищу данных.
- ▶ Формирование ответа (template)
- ▶ Отправка ответа клиенту.

Традиционная реализация

- ▶ один процесс на один цикл

Недоиспользование железа

Рассмотрим типичный вебсервер. на PHP, Perl, Ruby...

Задачи одного цикла

- ▶ Чтение запроса из сети.
- ▶ Парсер запроса http.
- ▶ Валидация запроса, выбор контроллера.
- ▶ Запросы к хранилищу данных.
- ▶ Формирование ответа (template)
- ▶ Отправка ответа клиенту.

Традиционная реализация

- ▶ один процесс на один цикл
- ▶ один тред на один цикл

Недоиспользование железа

Начались разговоры о HighLoad?

Недоиспользование железа

Начались разговоры о HighLoad?

- ▶ Увеличение числа процессов/тредов.

Недоиспользование железа

Начались разговоры о HighLoad?

- ▶ Увеличение числа процессов/тредов.
- ▶ Увеличение числа серверов.

Недоиспользование железа

Вернемся к рассматриваемому серверу

Недоиспользование железа

Вернемся к рассматриваемому серверу

- ▶ Проблемы наступили при 100 запросах в секунду.

Недоиспользование железа

Вернемся к рассматриваемому серверу

- ▶ Проблемы наступили при 100 запросах в секунду.
- ▶ Увеличили число процессов в работе.

Недоиспользование железа

Вернемся к рассматриваемому серверу

- ▶ Проблемы наступили при 100 запросах в секунду.
- ▶ Увеличили число процессов в работе.
- ▶ Новые проблемы при 150 запросов в секунду.

Недоиспользование железа

Что делать?

Недоиспользование железа

Что делать?

- ▶ Менять архитектуру?

Недоиспользование железа

Что делать?

- ▶ Менять архитектуру?
 - Мы над этим 3 года работали!

Недоиспользование железа

Что делать?

- ▶ Менять архитектуру?
 - Мы над этим 3 года работали!
- ▶ Добавлять второй сервер?

Недоиспользование железа

Что делать?

- ▶ Менять архитектуру?
 - Мы над этим 3 года работали!
- ▶ Добавлять второй сервер?
 - Это тоже не просто!
(бизнеслогика)

Спокойно!

Недоиспользование железа

Что делать?

- ▶ Менять архитектуру?
 - Мы над этим 3 года работали!
- ▶ Добавлять второй сервер?
 - Это тоже не просто!
(бизнеслогика)

Спокойно!

- ▶ Провести анализ архитектуры

Недоиспользование железа

Что делать?

- ▶ Менять архитектуру?
 - Мы над этим 3 года работали!
- ▶ Добавлять второй сервер?
 - Это тоже не просто!
(бизнеслогика)

Спокойно!

- ▶ Провести анализ архитектуры
- ▶ Провести измерения

Недоиспользование железа

Измерения: ищем узкое место?

Недоиспользование железа

Измерения: ищем узкое место?

Чтение запроса из сети.

15K RPS

Недоиспользование железа

Измерения: ищем узкое место?

Чтение запроса из сети.

15K RPS

Парсинг запроса, валидация, выбор контроллера.

150K RPS/CPU

Недоиспользование железа

Измерения: ищем узкое место?

Чтение запроса из сети.

Парсинг запроса, валидация, выбор контроллера.

Запросы к хранилищу.

15K RPS

150K RPS/CPU

60K RPS

Недоиспользование железа

Измерения: ищем узкое место?

Чтение запроса из сети.	15K RPS
Парсинг запроса, валидация, выбор контроллера.	150K RPS/CPU
Запросы к хранилищу.	60K RPS
Формирование ответа (Соединение данных с template)	100K RPS/CPU

Недоиспользование железа

Измерения: ищем узкое место?

Чтение запроса из сети.	15K RPS
Парсинг запроса, валидация, выбор контроллера.	150K RPS/CPU
Запросы к хранилищу.	60K RPS
Формирование ответа (Соединение данных с template)	100K RPS/CPU
Отправка ответа клиенту.	15K RPS

Недоиспользование железа

Итого

$$\frac{1}{\frac{1}{15 \cdot 10^3} + \frac{1}{150 \cdot 10^3} + \frac{1}{60 \cdot 10^3} + \frac{1}{100 \cdot 10^3} + \frac{1}{15 \cdot 10^3}} = 6000 RPS$$

Недоиспользование железа

Итого

$$\frac{1}{\frac{1}{15 \cdot 10^3} + \frac{1}{150 \cdot 10^3} + \frac{1}{60 \cdot 10^3} + \frac{1}{100 \cdot 10^3} + \frac{1}{15 \cdot 10^3}} = 6000 RPS$$

Но, позвольте!

Недоиспользование железа

Итого

$$\frac{1}{\frac{1}{15 \cdot 10^3} + \frac{1}{150 \cdot 10^3} + \frac{1}{60 \cdot 10^3} + \frac{1}{100 \cdot 10^3} + \frac{1}{15 \cdot 10^3}} = 6000 RPS$$

Но, позвольте!

- У нас проблемы на 150 RPS!

Недоиспользование железа

Итого

$$\frac{1}{\frac{1}{15 \cdot 10^3} + \frac{1}{150 \cdot 10^3} + \frac{1}{60 \cdot 10^3} + \frac{1}{100 \cdot 10^3} + \frac{1}{15 \cdot 10^3}} = 6000 RPS$$

Но, позвольте!

- ▶ У нас проблемы на 150 RPS!
- ▶ Тут что-то не так!

Недоиспользование железа

Начинаем разбираться

Недоиспользование железа

Начинаем разбираться

- ▶ Хранилище выходит на свои RPS при достаточно большом числе соединений к нему.

Недоиспользование железа

Начинаем разбираться

- ▶ Хранилище выходит на свои RPS при достаточно большом числе соединений к нему.
- ▶ То же самое и с взаимодействием с клиентом.

Недоиспользование железа

Начинаем разбираться

- ▶ Хранилище выходит на свои RPS при достаточно большом числе соединений к нему.
- ▶ То же самое и с взаимодействием с клиентом.
- ▶ Хранилище локально выдает еще больше RPS причем и на одном соединении!

Недоиспользование железа

Начинаем разбираться

- ▶ Хранилище выходит на свои RPS при достаточно большом числе соединений к нему.
- ▶ То же самое и с взаимодействием с клиентом.
- ▶ Хранилище локально выдает еще больше RPS причем и на одном соединении!

Что делать?

Недоиспользование железа

Начинаем разбираться

- ▶ Хранилище выходит на свои RPS при достаточно большом числе соединений к нему.
- ▶ То же самое и с взаимодействием с клиентом.
- ▶ Хранилище локально выдает еще больше RPS причем и на одном соединении!

Что делать?

- ▶ Попробуем еще увеличить число процессов?

Недоиспользование железа

Начинаем разбираться

- ▶ Хранилище выходит на свои RPS при достаточно большом числе соединений к нему.
- ▶ То же самое и с взаимодействием с клиентом.
- ▶ Хранилище локально выдает еще больше RPS причем и на одном соединении!

Что делать?

- ▶ Попробуем еще увеличить число процессов?
- ▶ Проблемы стали больше!

Недоиспользование железа

Начинаем разбираться

- ▶ Хранилище выходит на свои RPS при достаточно большом числе соединений к нему.
- ▶ То же самое и с взаимодействием с клиентом.
- ▶ Хранилище локально выдает еще больше RPS причем и на одном соединении!

Что делать?

- ▶ Попробуем еще увеличить число процессов?
- ▶ Проблемы стали больше!
- ▶ Почему?!

Недоиспользование железа

Резюме ситуации, еще раз

Недоиспользование железа

Резюме ситуации, еще раз

- ▶ Имеется 100500 строк кода, над которым работали несколько лет.

Недоиспользование железа

Резюме ситуации, еще раз

- ▶ Имеется 100500 строк кода, над которым работали несколько лет.
- ▶ Этот код AS IS по результатам измерений может выдавать гораздо больше RPS чем в реальности.

Недоиспользование железа

Резюме ситуации, еще раз

- ▶ Имеется 100500 строк кода, над которым работали несколько лет.
- ▶ Этот код AS IS по результатам измерений может выдавать гораздо больше RPS чем в реальности.
- ▶ Проблемы начинаются на уровне RPS на порядок меньшими нежели расчетные.

Недоиспользование железа

Еще раз рассмотрим цикл обработки

Недоиспользование железа

Еще раз рассмотрим цикл обработки

- ▶ Ожидание запроса от пользователя.

Недоиспользование железа

Еще раз рассмотрим цикл обработки

- ▶ Ожидание запроса от пользователя.
- ▶ Парсинг запроса, валидация.

Недоиспользование железа

Еще раз рассмотрим цикл обработки

- ▶ Ожидание запроса от пользователя.
- ▶ Парсинг запроса, валидация.
- ▶ Формирование запроса (запросов) в БД.

Недоиспользование железа

Еще раз рассмотрим цикл обработки

- ▶ Ожидание запроса от пользователя.
- ▶ Парсинг запроса, валидация.
- ▶ Формирование запроса (запросов) в БД.
- ▶ Ожидание ответа (ответов) из БД.

Недоиспользование железа

Еще раз рассмотрим цикл обработки

- ▶ Ожидание запроса от пользователя.
- ▶ Парсинг запроса, валидация.
- ▶ Формирование запроса (запросов) в БД.
- ▶ Ожидание ответа (ответов) из БД.
- ▶ Соединение данных из БД с шаблоном.

Недоиспользование железа

Еще раз рассмотрим цикл обработки

- ▶ Ожидание запроса от пользователя.
- ▶ Парсинг запроса, валидация.
- ▶ Формирование запроса (запросов) в БД.
- ▶ Ожидание ответа (ответов) из БД.
- ▶ Соединение данных из БД с шаблоном.
- ▶ Ожидание отправки данных клиенту.

Недоиспользование железа

Еще раз рассмотрим цикл обработки

- ▶ Ожидание запроса от пользователя.
- ▶ Парсинг запроса, валидация.
- ▶ Формирование запроса (запросов) в БД.
- ▶ Ожидание ответа (ответов) из БД.
- ▶ Соединение данных из БД с шаблоном.
- ▶ Ожидание отправки данных клиенту.
- ▶ Следующий клиент!

Недоиспользование железа

Измеряем

Недоиспользование железа

Измеряем

Ожидание запроса от пользователя. 70 мкс

Недоиспользование железа

Измеряем

Ожидание запроса от пользователя. 70 мкс

Парсинг запроса, валидация. 6 мкс

Недоиспользование железа

Измеряем

Ожидание запроса от пользователя.	70 мкс
Парсинг запроса, валидация.	6 мкс
Формирование запроса (запросов) в БД.	1 мкс

Недоиспользование железа

Измеряем

Ожидание запроса от пользователя.	70 мкс
Парсинг запроса, валидация.	6 мкс
Формирование запроса (запросов) в БД.	1 мкс
Ожидание ответа (ответов) из БД.	16 мкс

Недоиспользование железа

Измеряем

Ожидание запроса от пользователя.	70 мкс
Парсинг запроса, валидация.	6 мкс
Формирование запроса (запросов) в БД.	1 мкс
Ожидание ответа (ответов) из БД.	16 мкс
Соединение данных из БД с шаблоном.	10 мкс

Недоиспользование железа

Измеряем

Ожидание запроса от пользователя.	70 мкс
Парсинг запроса, валидация.	6 мкс
Формирование запроса (запросов) в БД.	1 мкс
Ожидание ответа (ответов) из БД.	16 мкс
Соединение данных из БД с шаблоном.	10 мкс
Ожидание отправки данных клиенту.	70 мкс

Недоиспользование железа

Итого

- ▶ Код выполнялся: $6 + 1 + 10 =$

Недоиспользование железа

Итого

- ▶ Код выполнялся: $6 + 1 + 10 = 17$ мкс

Недоиспользование железа

Итого

- ▶ Код выполнялся: $6 + 1 + 10 = 17$ мкс
- ▶ Чего-либо ожидали: $70 + 16 + 70 =$

Недоиспользование железа

Итого

- ▶ Код выполнялся: $6 + 1 + 10 = 17$ мкс
- ▶ Чего-либо ожидали: $70 + 16 + 70 = 156$ мкс

Недоиспользование железа

Итого

- ▶ Код выполнялся: $6 + 1 + 10 = 17$ мкс
- ▶ Чего-либо ожидали: $70 + 16 + 70 = 156$ мкс
- ▶ Наш код выполняется только 10% времени!

Недоиспользование железа

Итого

- ▶ Код выполнялся: $6 + 1 + 10 = 17$ мкс
- ▶ Чего-либо ожидали: $70 + 16 + 70 = 156$ мкс
- ▶ Наш код выполняется только 10% времени!
- ▶ И при этом тормозит!

Недоиспользование железа

```
#include <unistd.h>

int main(int argc, char **argv) {
    int i;
    for (;;) {
        usleep(70);      usleep(7);
        usleep(16);      usleep(10);
        usleep(70);
    }
}
```


Недоиспользование железа

Итого

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU
- по моим измерениям - где-то 15% загрузки на CPU

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU
- по моим измерениям - где-то 15% загрузки на CPU
- ▶ Запустив десяток таких “воркеров”, получаем примерно такую же нагрузку как на проблемном сервере.

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU
- по моим измерениям - где-то 15% загрузки на CPU
- ▶ Запустив десяток таких “воркеров”, получаем примерно такую же нагрузку как на проблемном сервере.
- ▶ Понятно что пример синтетический (есть вопросы к реализации `usleep`).

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU
- по моим измерениям - где-то 15% загрузки на CPU
- ▶ Запустив десяток таких “воркеров”, получаем примерно такую же нагрузку как на проблемном сервере.
- ▶ Понятно что пример синтетический (есть вопросы к реализации `usleep`).

Вернемся к нашему серверу

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU
- по моим измерениям - где-то 15% загрузки на CPU
- ▶ Запустив десяток таких “воркеров”, получаем примерно такую же нагрузку как на проблемном сервере.
- ▶ Понятно что пример синтетический (есть вопросы к реализации `usleep`).

Вернемся к нашему серверу

- ▶ Каждая отдельная часть имеет хорошую производительность

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU - по моим измерениям - где-то 15% загрузки на CPU
- ▶ Запустив десяток таких “воркеров”, получаем примерно такую же нагрузку как на проблемном сервере.
- ▶ Понятно что пример синтетический (есть вопросы к реализации `usleep`).

Вернемся к нашему серверу

- ▶ Каждая отдельная часть имеет хорошую производительность достаточную для развития проекта еще на несколько лет вперед.

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU - по моим измерениям - где-то 15% загрузки на CPU
- ▶ Запустив десяток таких “воркеров”, получаем примерно такую же нагрузку как на проблемном сервере.
- ▶ Понятно что пример синтетический (есть вопросы к реализации `usleep`).

Вернемся к нашему серверу

- ▶ Каждая отдельная часть имеет хорошую производительность достаточную для развития проекта еще на несколько лет вперед.
- ▶ Большую часть времени (90%) наш код проводит в ожидании.

Недоиспользование железа

Итого

- ▶ Код, делающий только `sleep` в цикле неплохо грузит CPU - по моим измерениям - где-то 15% загрузки на CPU
- ▶ Запустив десяток таких “воркеров”, получаем примерно такую же нагрузку как на проблемном сервере.
- ▶ Понятно что пример синтетический (есть вопросы к реализации `usleep`).

Вернемся к нашему серверу

- ▶ Каждая отдельная часть имеет хорошую производительность достаточную для развития проекта еще на несколько лет вперед.
- ▶ Большую часть времени (90%) наш код проводит в ожидании.
- ▶ Что делать?

Недоиспользование железа

Просто реорганизовать код

Событийно-ориентированное программирование

Компьютер — это конечный автомат. Труды для тех людей, которые не умеют программировать конечные автоматы.

Алан Кокс

Событийно-ориентированное программирование

Избавимся от тредов!

Событийно-ориентированное программирование

Событийно-ориентированное программирование

Событийно-ориентированное программирование

Событийно-ориентированное программирование

Событийно-ориентированное программирование

Событийно-ориентированное программирование