



redis

«REDIS:

ОСНОВЫ И ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ»

ОБРАЩЕНИЕ К «СКЛАДЧИКАМ» И ПОЛЬЗОВАТЕЛЯМ ТОРРЕНТОВ

Я считаю складчины, а тем более видеоролики, скачанные с торрент-трекеров – полной ерундой. Невозможно научиться чему-то, просматривая видео. Без домашних заданий, без общения с преподавателями и коллегами. Пользуясь пиратскими видеозаписями курсов, вы вредите прежде всего самим себе, создавая иллюзию «обучения». И поддерживаете непонятных «организаторов», имеющих свою прибыль с подобных сервисов.

Впрочем, дело ваше.

Однако, если вы хотите по-настоящему учиться – приходите. Адрес есть на слайдах.



О чём вообще сегодняшний вебинар?

- С одной стороны у нас есть традиционные реляционные ACID базы данных: MySQL, Postgres, Oracle и другие.
- Они надежны и стабильны, сама аббревиатура ACID говорит об этом!
- С другой же стороны есть очень-очень быстрые хранилища-в-памяти, например memcache
- Они быстрые за счет предельной простоты и отказа от надежности
- А нет ли чего, что было бы посередине?
- Чтобы было и быстро (от 100K/s), и сложные структуры (а не просто ключ-значение), и чтобы минимальная надежность хранения, а еще бы масштабирование?

Есть – Redis!

О ЧЕМ РАЗГОВОР?



УСТАНОВКА И НАЧАЛО РАБОТЫ

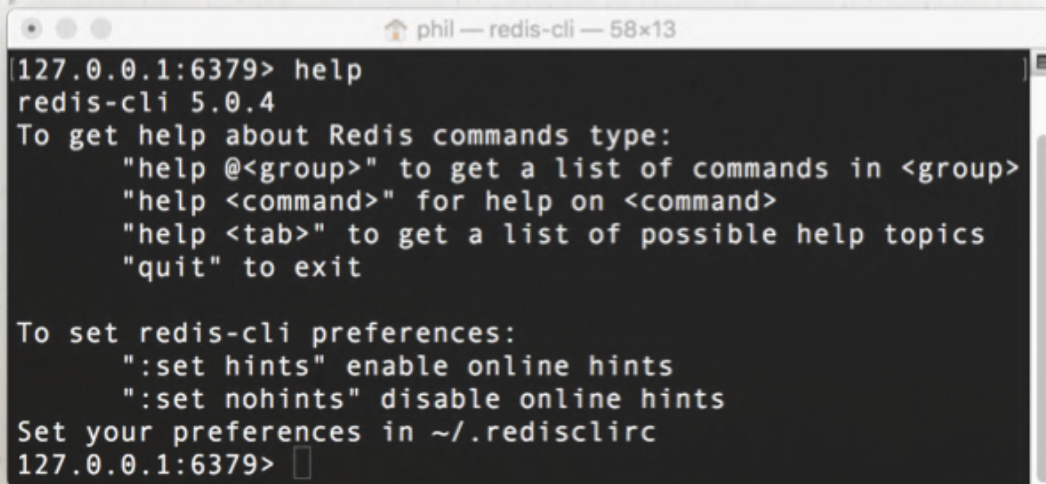


redis

Redis:

Remote Dictionary Server

- Redis распространяется по лицензии BSD
- Написан на языке C, работает на Linux и macOS
- Официальной версии для Windows нет, но есть в составе Open Server (старая версия 3.2)
- Однако у нас всегда есть Docker ☺
- Последняя на данный момент стабильная версия: 5.0.4
- Библиотеки для работы с Redis есть во всех современных языках программирования



```
phil — redis-cli — 58x13
127.0.0.1:6379> help
redis-cli 5.0.4
To get help about Redis commands type:
  "help @<group>" to get a list of commands in <group>
  "help <command>" for help on <command>
  "help <tab>" to get a list of possible help topics
  "quit" to exit

To set redis-cli preferences:
  ":set hints" enable online hints
  ":set nohints" disable online hints
Set your preferences in ~/.redisclirc
127.0.0.1:6379> 
```

НАЧАЛО
РАБОТЫ

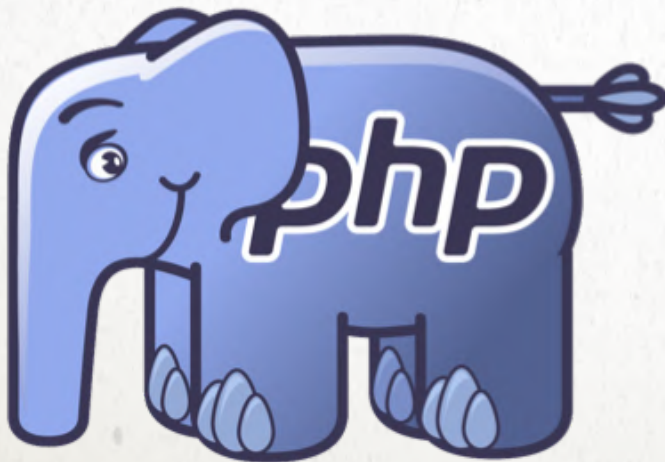


Redis:

Remote Dictionary Server

- Библиотеки для работы с Redis есть во всех современных языках программирования
- Для **PHP** я рекомендую использовать phpredis – это бинарное расширение
- Самый простой способ установки – через PECL

```
$redis = new Redis();  
$redis->connect('127.0.0.1', 6379);
```



НАЧАЛО
РАБОТЫ



ИСПОЛЬЗОВАНИЕ КАК «КЛЮЧ-ЗНАЧЕНИЕ»



redis

Redis:

«КЛЮЧ-ЗНАЧЕНИЕ»

- Самая простая операция – это установка строкового значения по ключу и чтение:

```
127.0.0.1:6379> EXISTS foo  
(integer) 0
```

```
127.0.0.1:6379> SET foo "Hello!"  
OK
```

```
127.0.0.1:6379> GET foo  
"Hello!"
```

```
127.0.0.1:6379> EXISTS foo  
(integer) 1
```

- Чуть сложнее вариант с установкой срока жизни значения в секундах или миллисекундах:

```
127.0.0.1:6379> SET foo "Hello!" EX 5  
OK
```

```
127.0.0.1:6379> GET foo  
"Hello!"
```

```
127.0.0.1:6379> GET foo  
(nil)
```

ПРОСТЫЕ ОПЕРАЦИИ



Redis:

«КЛЮЧ-ЗНАЧЕНИЕ»

- Установка нового значения с возвращением старого:

```
127.0.0.1:6379> SET foo "Hello!"  
OK
```

```
127.0.0.1:6379> GETSET foo "World"  
"Hello!"
```

```
127.0.0.1:6379> GET foo  
"World"
```

- Добавление к значению строки:

```
127.0.0.1:6379> SET foo "Hello"  
OK
```

```
127.0.0.1:6379> APPEND foo " World"  
(integer) 11
```

```
127.0.0.1:6379> GET foo  
"Hello World"
```

ПРОСТЫЕ ОПЕРАЦИИ



Redis:

«КЛЮЧ-ЗНАЧЕНИЕ»

- Инкремент и декремент (счетчики)

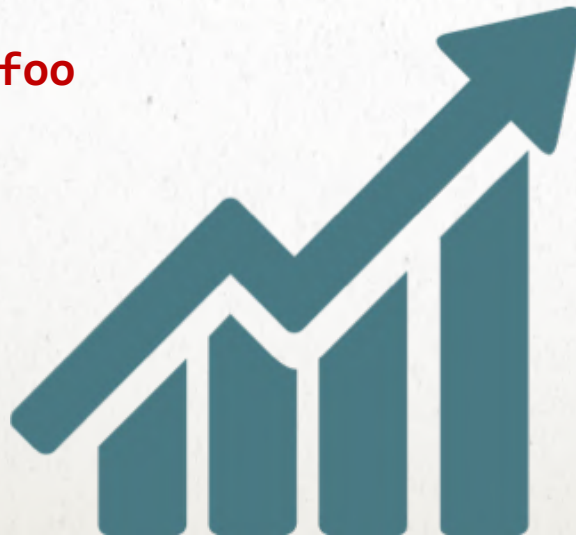
```
127.0.0.1:6379> SET foo 1  
OK
```

```
127.0.0.1:6379> INCR foo  
(integer) 2
```

```
127.0.0.1:6379> GET foo  
"2"
```

```
127.0.0.1:6379> DECR foo  
(integer) 1
```

```
127.0.0.1:6379> GET foo  
"1"
```



ПРОСТЫЕ ОПЕРАЦИИ



КОМПЛЕКСНЫЕ ТИПЫ ДАННЫХ



redis

Redis:

«Хэш-таблицы»

- Хэш-таблицу можно представлять себе как объект, состоящий из полей. Каждое поле имеет своё имя и значение.
- Другое название – «ассоциативный массив»

```
127.0.0.1:6379> HSET obj foo 42  
(integer) 1
```

```
127.0.0.1:6379> HGET obj foo  
"42"
```

```
127.0.0.1:6379> HSET obj bar 24  
(integer) 1
```

```
127.0.0.1:6379> HGETALL obj
```

```
1) "foo"  
2) "42"  
3) "bar"  
4) "24"
```

```
127.0.0.1:6379> HVALS obj
```

```
1) "42"  
2) "24"
```

HASHES



Redis:

«Множество»

- Множество – это неупорядоченный набор значений.

```
127.0.0.1:6379> SADD myset "Hello"  
(integer) 1
```

```
127.0.0.1:6379> SADD myset "World"  
(integer) 1
```

```
127.0.0.1:6379> SMEMBERS myset  
1) "World"  
2) "Hello"
```

```
127.0.0.1:6379> SCARD myset  
(integer) 2
```

SISMEMBER, UNION,
SDIFF, SINTER,
SREM, SPOP, SMOVE



SETS



Redis:

«Списки»

- Список – это последовательность значений, упорядоченных по порядку их создания.

```
127.0.0.1:6379> LPUSH mylist "first"  
(integer) 1
```

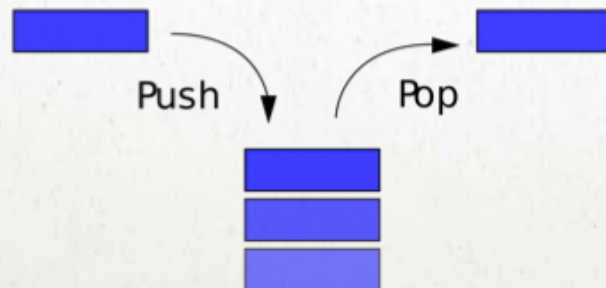
```
127.0.0.1:6379> LPUSH mylist "second"  
(integer) 2
```

```
127.0.0.1:6379> LPOP mylist  
"second"
```

```
127.0.0.1:6379>  
LINSERT mylist BEFORE first "zero"  
(integer) 2
```

```
127.0.0.1:6379> LRANGE mylist 0 -1
```

```
1) "zero"  
2) "first"
```



LISTS



Redis:

«Упорядоченные множества»

А еще есть упорядоченные множества. Это как множество, но вы можете задать порядок 😊

```
127.0.0.1:6379> ZADD myzset 2 "one"  
(integer) 1
```

```
127.0.0.1:6379> ZADD myzset 1 "two"  
(integer) 1
```

```
127.0.0.1:6379>  
ZRANGE myzset 0 -1 WITHSCORES
```

- 1) "two"
- 2) "1"
- 3) "one"
- 4) "2"



ZSETS



ТРАНЗАКЦИИ В REDIS



redis

Redis:

«Транзакции»

Транзакция: множество команд, выполняющихся, как одна команда.

```
127.0.0.1:6379> MULTI  
OK
```

```
127.0.0.1:6379> INCR foo  
QUEUED
```

```
127.0.0.1:6379> INCR bar  
QUEUED
```

```
127.0.0.1:6379> EXEC  
1) (integer) 3  
2) (integer) 1
```

Команда **DISCARD** прерывает набор команд в транзакцию.

Н. В. Важно понимать, что в Redis нет отката! Сбой одной из команд в «транзакции» не приводит к отмене предыдущих, вместо этого выполняются все следующие ☹

ТРАНЗАКЦИИ



МЕХАНИЗМ ПОДПИСОК "PUB-SUB"



redis

Redis:

«Подписки»

Механизм подписок в Redis устроен очень просто.

- Одни клиенты сервиса Redis подписываются на определенные каналы сообщений:

```
127.0.0.1:6379> SUBSCRIBE channel1
```

Reading messages... (press Ctrl-C to quit)

```
1) "subscribe"  
2) "channel1"  
3) (integer) 1
```

- Другие клиенты могут посылать сообщения в каналы. Подписанные на данные каналы клиенты получат сообщения:

```
127.0.0.1:6379>  
PUBLISH channel1 "Hello!"  
(integer) 1
```

```
...  
3) "Hello!"
```

PUB-SUB



ПРАКТИКА ПРИМЕНЕНИЯ REDIS



redis

Redis:

Практика применения

Redis достаточно широко применяется в современной разработке ПО.

Поддерживаемые языки программирования включают Java, Python, PHP, C, C++, C#, JavaScript, Node.js, Ruby, R, Go и многие другие.

Асинхронная репликация позволяет почти неограниченно масштабировать Redis.

- PHP может хранить данные сессий в Redis. Это дает нам увеличение производительности и отсутствие блокировок.
- Кэширование данных.
- Чаты и системы обмена сообщениями.
- Различные очереди задач
- Мгновенно сортируемые «таблицы лидеров»
- Мой личный опыт: мониторинг процессов

ЗАЧЕМ НУЖЕН?



ВМЕСТО ЗАКЛЮЧЕНИЯ

**ЗАПИСЫВАЙТЕСЬ
НА ВЕБИНАРЫ И КУРСЫ
АКАДЕМИИ ПРОГРАММИРОВАНИЯ**

