

Разработка высоконагруженных и надежных систем

Андрей Смирнов, 2015





Практическое задание №2

“Сравним” Redis и PostgreSQL

```
CREATE TABLE data (  
    id varchar NOT NULL,  
    f0 varchar,  
    f1 varchar,  
    f2 varchar,  
    f3 varchar,  
    f4 varchar,  
    f5 varchar,  
    f6 varchar,  
    f7 varchar,  
    f8 varchar,  
    f9 varchar,  
    PRIMARY KEY (id)  
);
```

```
CREATE OR REPLACE FUNCTION upsert_data_f0(key varchar, value varchar)
RETURNS VOID AS
$$
BEGIN
    LOOP
        UPDATE data SET f0 = value WHERE id = key;
        IF found THEN
            RETURN;
        END IF;
        BEGIN
            INSERT INTO data(id,f0) VALUES (key, value);
            RETURN;
        EXCEPTION WHEN unique_violation THEN
            END;
    END LOOP;
END;
$$
LANGUAGE plpgsql;
```

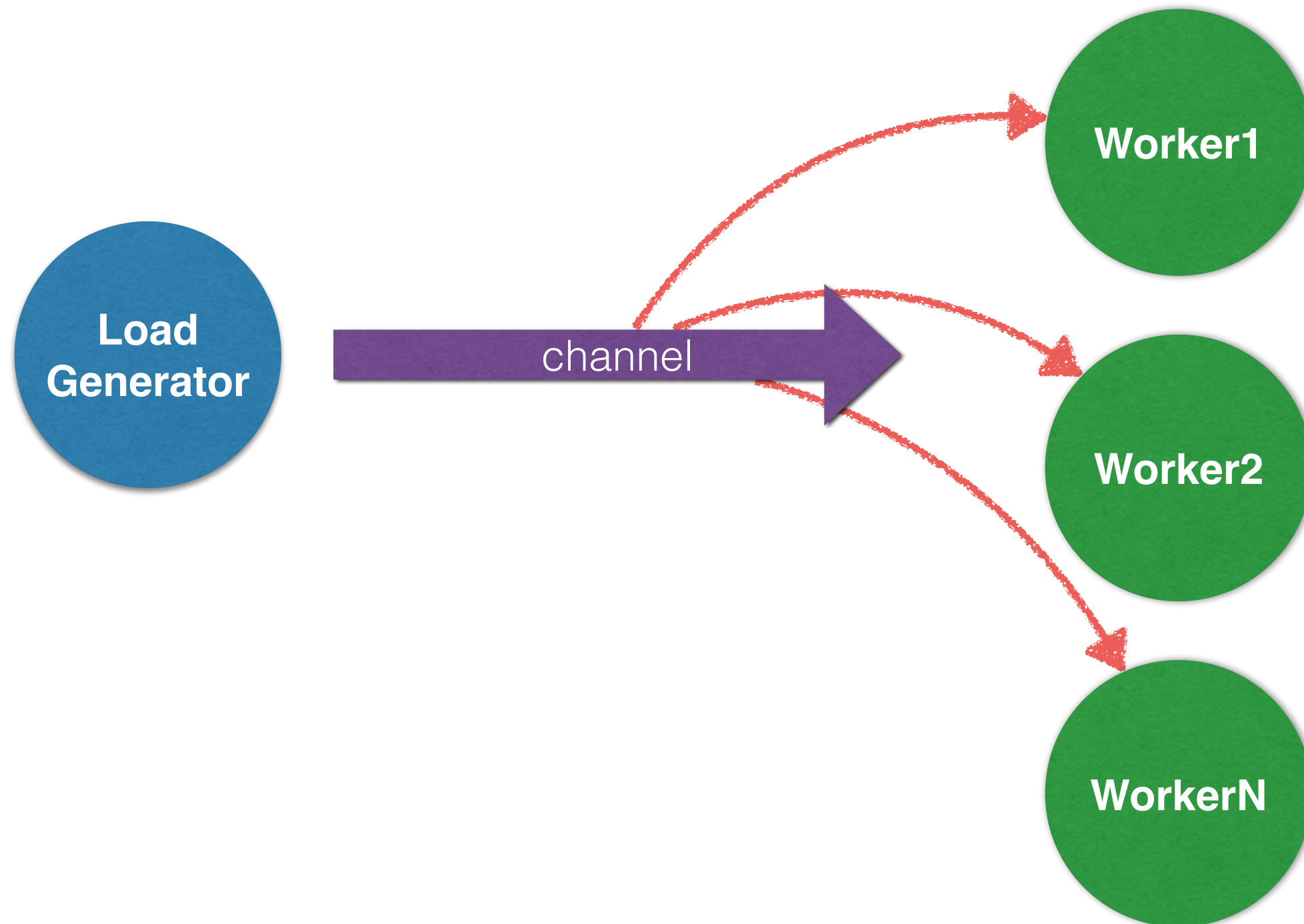
PostgreSQL

- `SELECT upsert_data_f3('obj123', 'x123.3')`

Redis

- HMSET obj123 f3 xx123.3

Внутренности теста



Первая фаза

```
for i := 0; i < numObjects && !stop; i++ {  
    for j := 0; j < 10 && !stop; j++ {  
        queue <- &Task{Op: OpWrite,  
                        Key: fmt.Sprintf("obj%d", i),  
                        Field: fmt.Sprintf("f%d", j),  
                        Value: fmt.Sprintf("xx%d.%d", i, j) }  
    }  
}
```


Вторая фаза

```
for !stop {  
    op := OpRead  
    i, j := rand.Int31n(int32(numObjects)), rand.Int31n(10)  
    if int(rand.Int31n(100)) < writePercent {  
        op = OpWrite  
    }  
    queue <- &Task{Op: op,  
        Key: fmt.Sprintf("obj%d", i),  
        Field: fmt.Sprintf("f%d", j),  
        Value: fmt.Sprintf("%d.%d", i, j) }  
}
```

```
$ docker run -t -i smira/hl-tasks:pgredis
hl-tasks@e236d7e042e1:~$ ./main -h
Usage of ./main:
  -mode="redis": operation mode: redis|postgres
  -num_objects=10000: number of objects
  -workers=4: number of parallel workers
  -write_percent=30: percent of write operations in rw phase

~/hl-tasks$ ./main -mode=redis
3 req/sec
4 req/sec
Going r/w
10 req/sec
^C Stopping...

~/hl-tasks$ ./main -mode=postgres
10 req/sec
11 req/sec
Going r/w
3 req/s
^C Stopping...
```



Почему PostgreSQL настолько
“медленнее” Redis? Чем это
обусловлено?

Честное ли было сравнение?



Можно ли ускорить первую фазу
(запись) при работе на
PostgreSQL?

Hint: “борьба” за одни строки в БД

пересобрать main:
`go build main.go`