



Scaling Redis with Python

Armin '@mitsuhiko' Ronacher



rb the
redis
blaster

- Distribute your dataset across many redis servers
- Scale to many times your RAM
- Still do this conveniently

- Auto-pipelining
- Auto-command batching (GET -> MGET)

```
$ pip install rb
```

```
from rb import Cluster

cluster = Cluster(hosts={
    0: {'port': 6379},
    1: {'port': 6380},
    2: {'port': 6381},
    3: {'port': 6382},
}, host_defaults={
    'host': '127.0.0.1',
})
```

Routing Client

```
results = {}  
client = cluster.get_routing_client()  
for key in keys_to_look_up:  
    results[key] = client.get(key)
```

Mapping Client

```
results = {}  
with cluster.map() as client:  
    for key in keys_to_look_up:  
        results[key] = client.get(key)
```

```
for key, promise in results.iteritems():  
    print '%s: %s' % (key, promise.value)
```


Fanout Client

```
with cluster.all() as client:  
    client.flushdb()
```

Fanout Client

```
with cluster.fanout(host_ids) as client:  
    result = client.info()  
  
for host_id, info in result.value.iteritems():  
    print '%d: %s' % (host_id, info['os'])
```

<http://rb.rtf.d.org/>



rb the
redis
blaster