



北京 上海 珠海 苏州 杭州

主办: PyChina.org, GDG

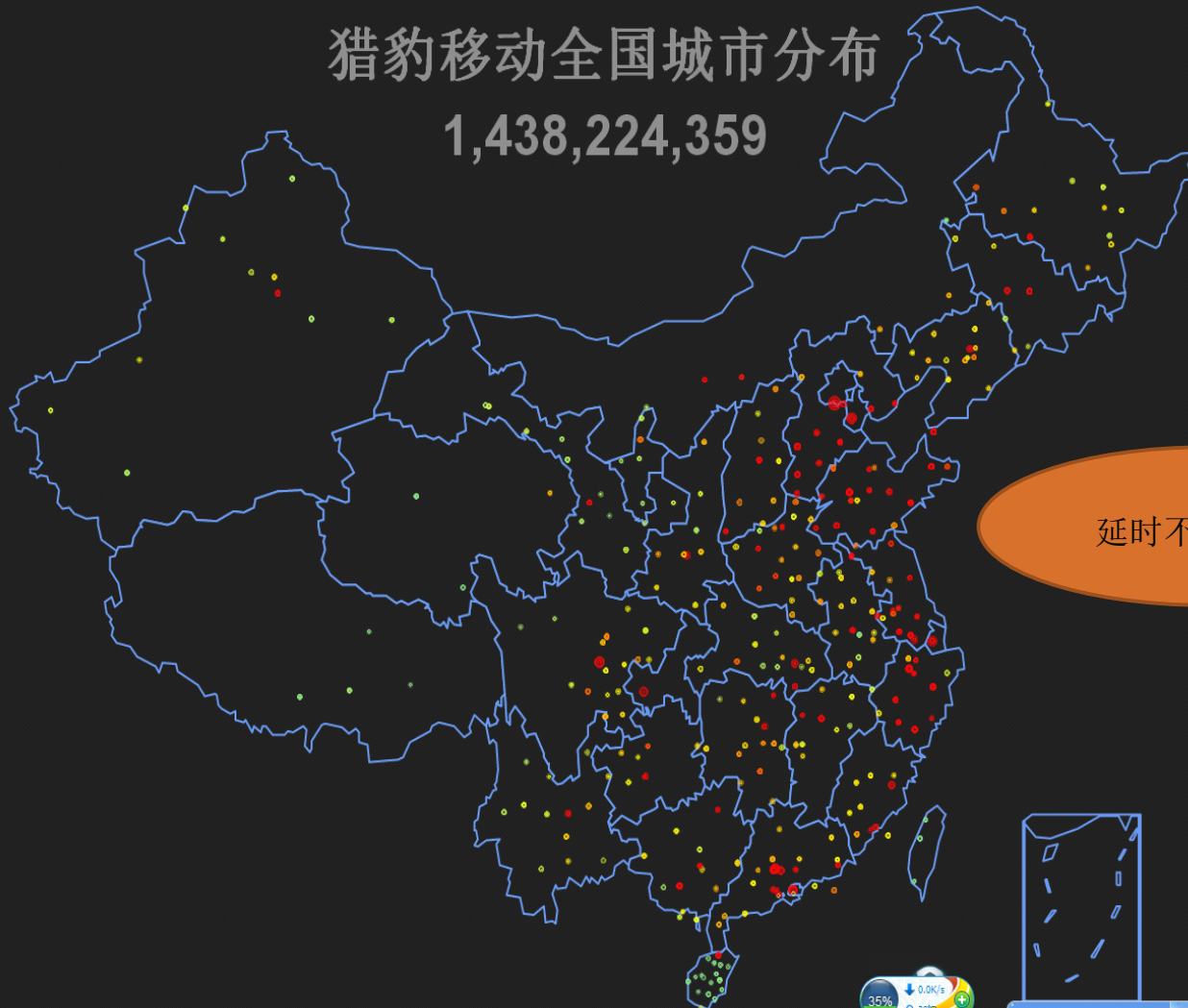
Luna之高效KV库

龙利民

- 北京市 10,163,855
- 天津市 6,530,105
- 成都市 5,794,995
- 上海市 5,720,505
- 广州市 5,608,410
- 深圳市 5,488,165
- 重庆市 4,341,370
- 杭州市 4,029,815
- 苏州市 2,903,660
- 温州市 2,885,655

猎豹移动全国城市分布

1,438,224,359



延时不超过5秒



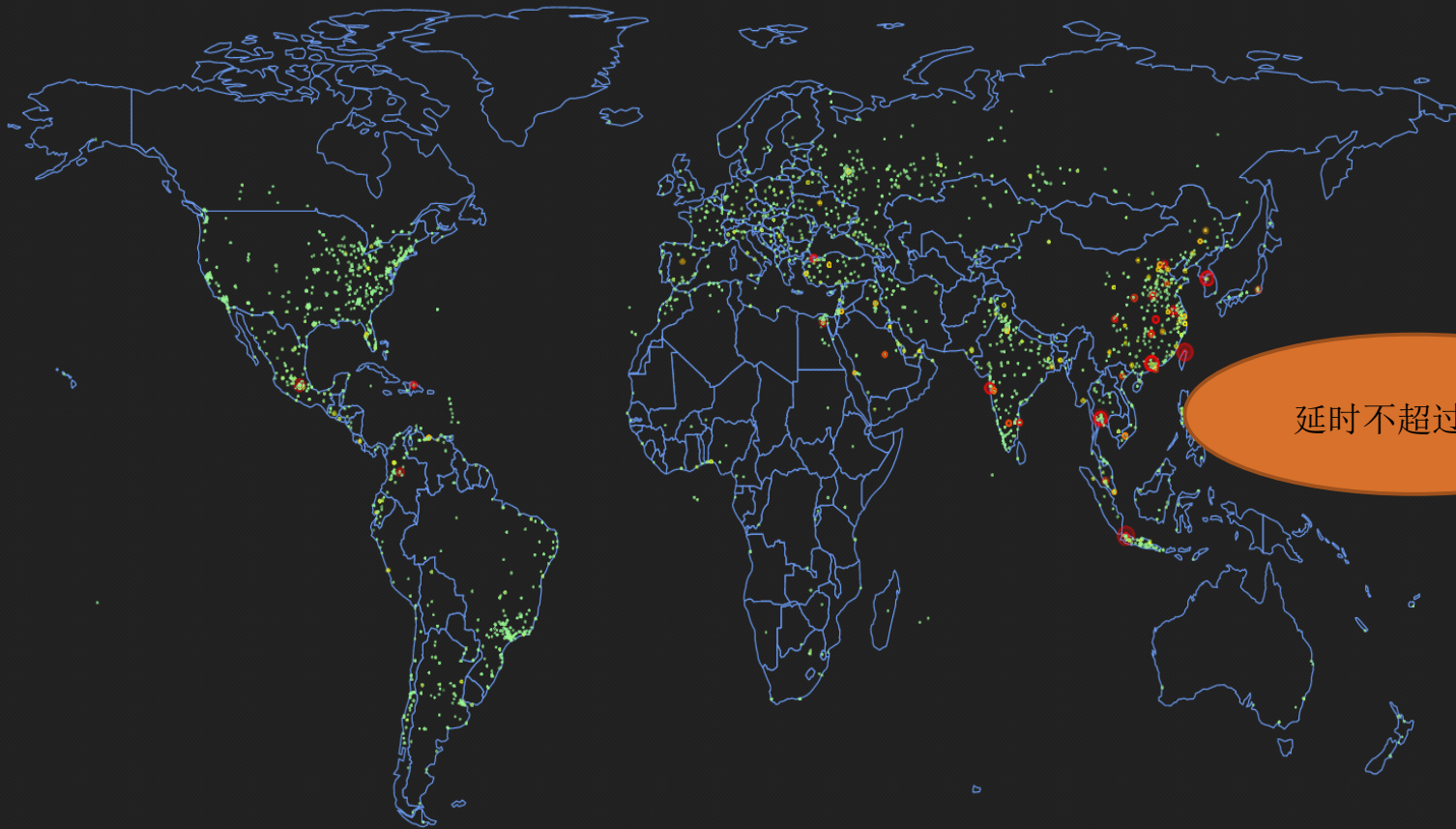
System status bar at the bottom right showing network speed (18.6KB/S), upload speed (2.1KB/S), and other indicators.



猎豹移动全球城市分布

3,548,148,779

- Jakarta 7,363,132
- Bangkok 7,307,728
- Taipei 6,839,511
- Guangzhou 5,970,150
- Seoul 5,394,350
- Mexico 4,636,520
- Nanjing 4,034,320
- Mumbai 3,836,224
- Zhengzhou 3,557,864
- Beijing 3,068,435

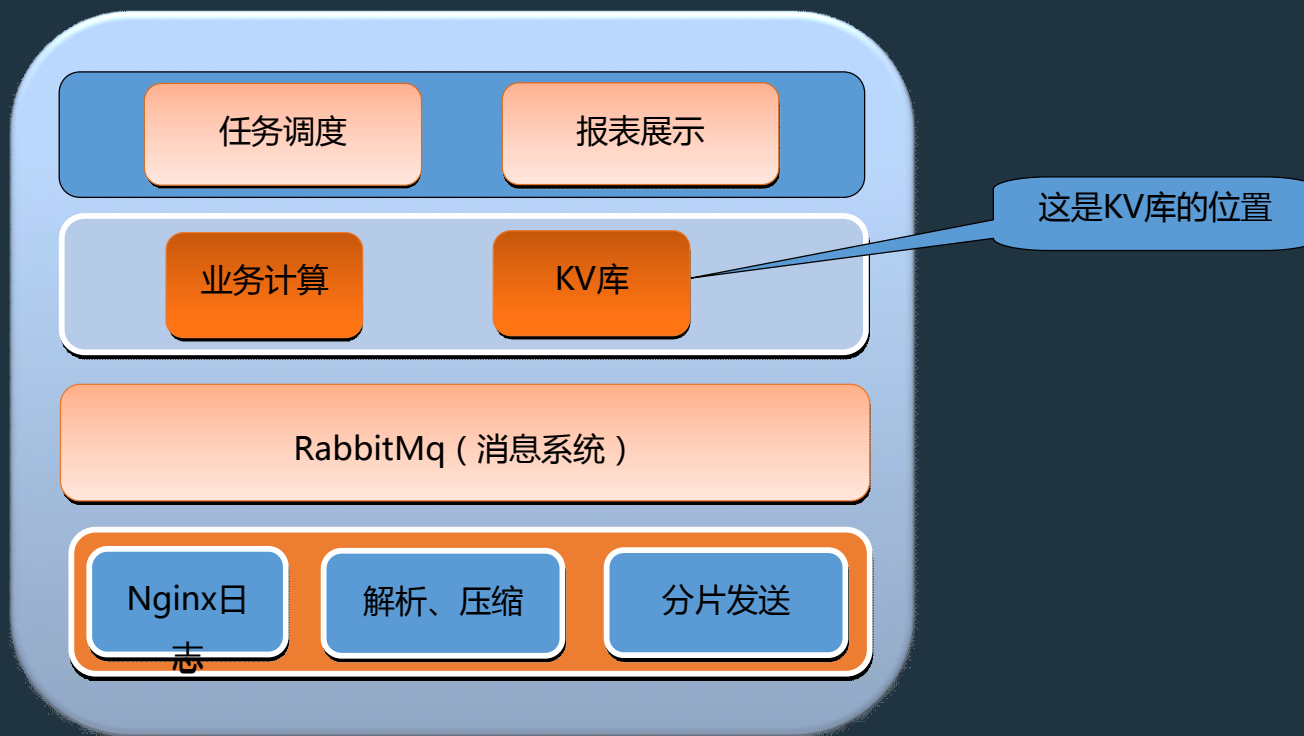


延时不超过20秒

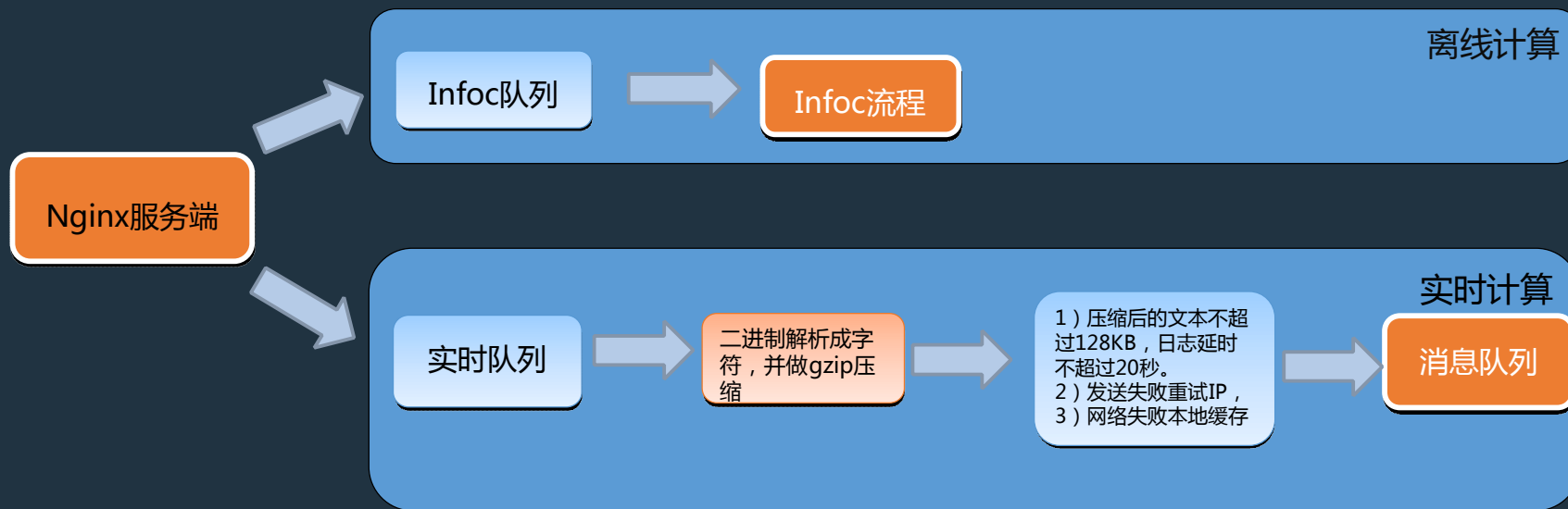


38% 0.0KB/S 0.0KB/S 47%

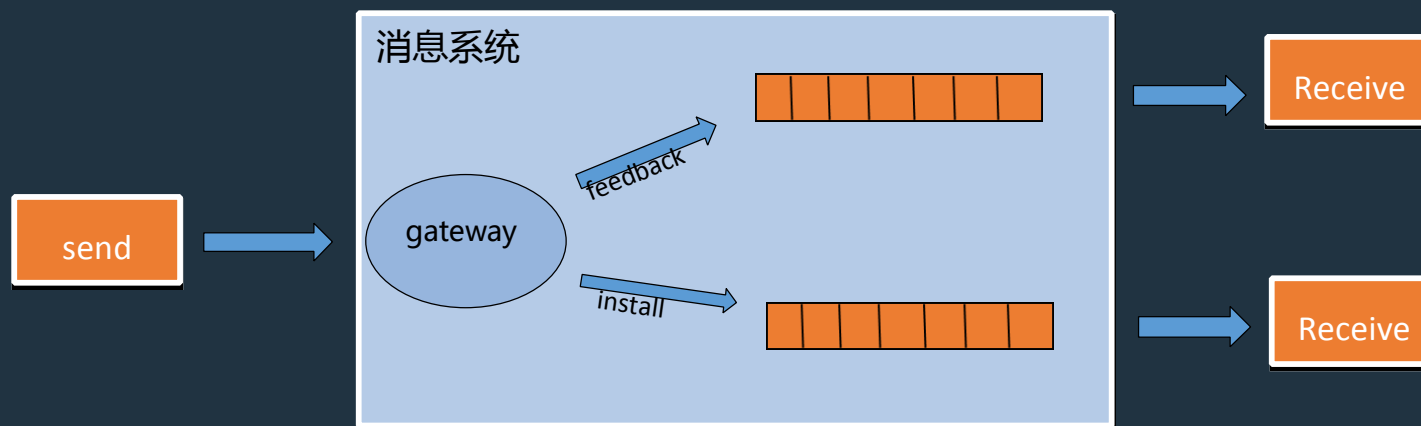
体系结构



消息发送



消息系统



消息的队列名是数据所对应的埋点名。每条消息包含有头部数据。

KV库演变

程序直接加载文件到内存，将KV库变量化

用关系型数据库存储KV

Leveldb本地缓存+Mysql(handlesocket)

LunaProxy+SSDB高性能数据库

高性能KV库遇到的最大的困难？

IO

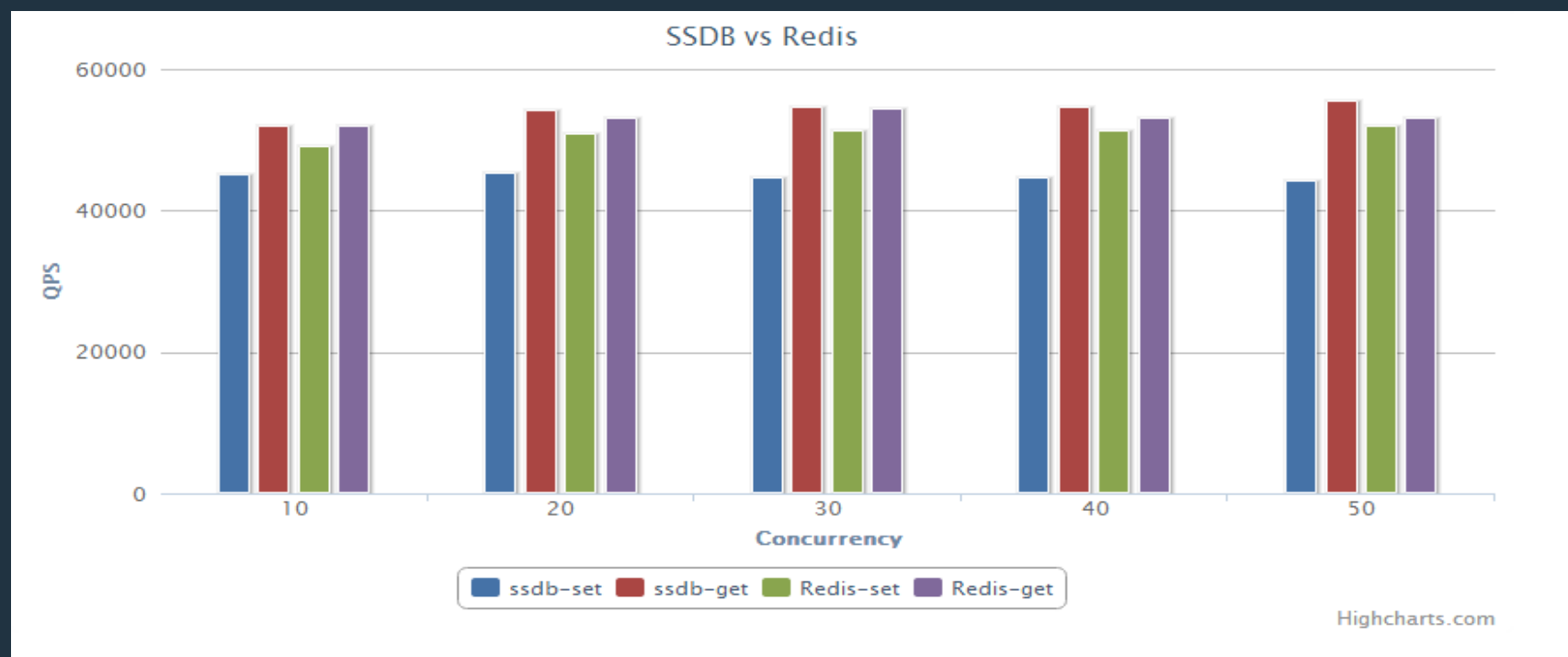
内存

第二问题是什么？

稳定性

SSDB

SSDB性能



解决IO和稳定性

ssdb.conf

```
work_dir = /dev/shm/ssdb/master1
pidfile = /dev/shm/ssdb/master1/ssdb.pid

server:
  ip: 172.28.26.178
  port: 8888

replication:
  binlog: yes
  # Limit sync speed to *MB/s, -1: no limit
  sync_speed: -1
  slaveof:
    type: sync
    ip: 172.28.26.175
    port: 8889

leveldb:
  cache_size: 500
  block_size: 32
  write_buffer_size: 64
  compaction_speed: 1000
  compression: yes
```

ssdb_slave.conf

```
work_dir = /data/ssdb/slave1
pidfile = /data/ssdb/slave1/ssdb.pid

server:
  ip: 172.28.26.175
  port: 8889

replication:
  binlog: yes
  sync_speed: 100
  slaveof:
    id: svc_1
    type: sync
    ip: 172.28.26.178
    port: 8888

leveldb:
  cache_size: 500
  block_size: 32
  write_buffer_size: 64
  compaction_speed: 200
  compression: yes
```

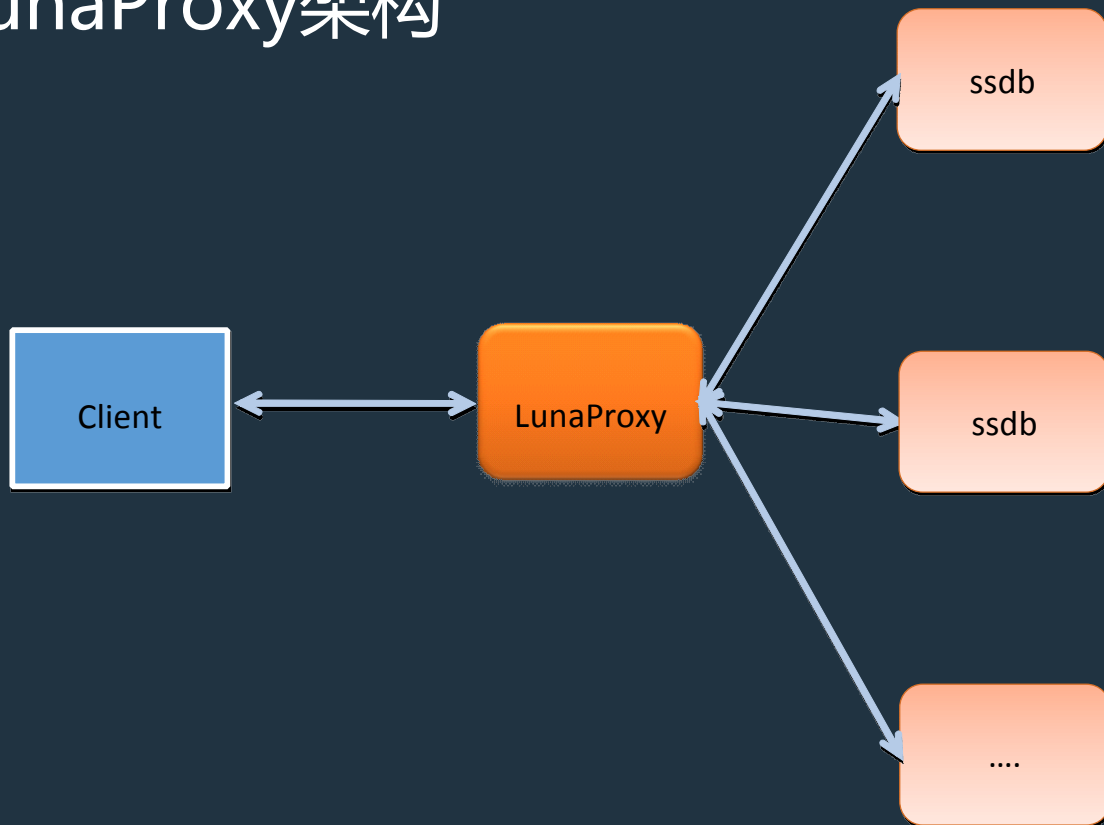
SSDB解决了什么问题

- “IO”（启用内存盘）
- 稳定性
- 跨业务逻辑访问

SSDB哪些问题解决不了？

- 线性扩展（加服务器能不能增加性能）
- 配置问题（实例多了，不好维护）
- 监控报警

LunaProxy架构



LunaProxy作用

- 通过对查询key的分片做到了线性扩展
- 业务需求可以配置
- 监控SSDB运行状态，出现异常及时报警

LunaProxy核心代码

➤SSDB可配置

```
HOST_PORT_CONF = {"ball_click":{"status":"split", "0":("192.28.26.100", "10000"), "1":("192.28.26.100", "10001"), "2":("192.28.26.100", "10002"), "f":("192.28.26.100", "10016")},  
                  "uninstall":{"status":"nosplit", "db":("192.28.26.200", "10000")}}
```

#真实的运行环境中用zookeeper实现配置管理

```
def is_split(name):
```

```
    return True if HOST_PORT_CONF[name]["status"] == "split" else False
```

```
def get_host_port(name, split_key=None):
```

```
    if split_key:
```

```
        return HOST_PORT_CONF[name][split_key]
```

```
    return HOST_PORT_CONF[name]["db"]
```

LunaProxy核心代码

➤线性扩展

```
def __multi_run(name, kset, func):
    results = []
    kmap = {}
    for key in kset:
        if key[0] not in kmap:
            kmap[key[0]] = set([])
        kmap[key[0]].append(key)
    i, threadpool = 0, []
    for split_key, hset in kmap.iteritems():
        results.insert(i, None)
        tt = threading.Thread(target=func, args=(name, split_key, hset, results[i]))
        tt.start()
        threadpool.append(tt)
    i = i + 1
    for i in xrange(0, len(kmap), 1):
        threadpool[i].join()
    return results
```

总结

- 用开源的、高性能数据库SSDB做为KV库；
- 内存盘作为SSDB的主库，提升IO性能；
- LunaProxy中通过对Key的切分实现线性扩展；
- 通过LunaProxy实现对SSDB的可配置化；
- 通过zookeeper做心跳检测，和实现故障转移。

Q&A