SDCC 2017深圳站
互联网应用架构实战峰会

CSDN

去内存化的缓存系统——Lest

阅文集团——帅翔

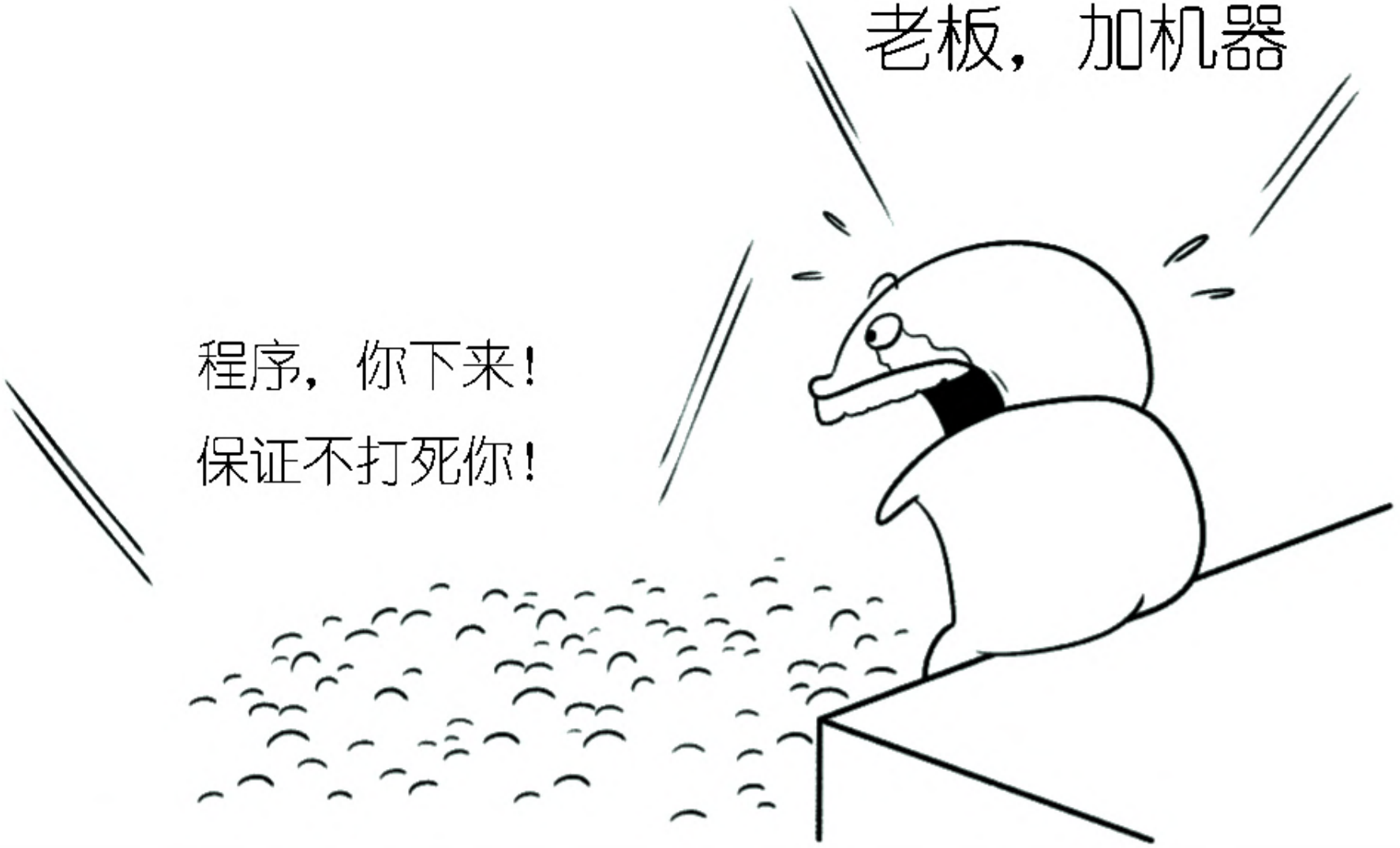

数字阅读
中国引领行业
和文学 IP 培
阅文集团
CHINA READING

帅 翔

毕业于上海大学计算机专业，目前就职于阅文集团内容中心，主要负责分布式文件系统的研发工作。关注分布式系统，数据存储，网络编程，linux系统，机器学习等相关领域。

主要内容

- 设计目标
- 总体架构
- Tracker服务
- 数据存储
- 数据同步
- 性能测试


老板，加机器
程序，你下来！
保证不打死你！

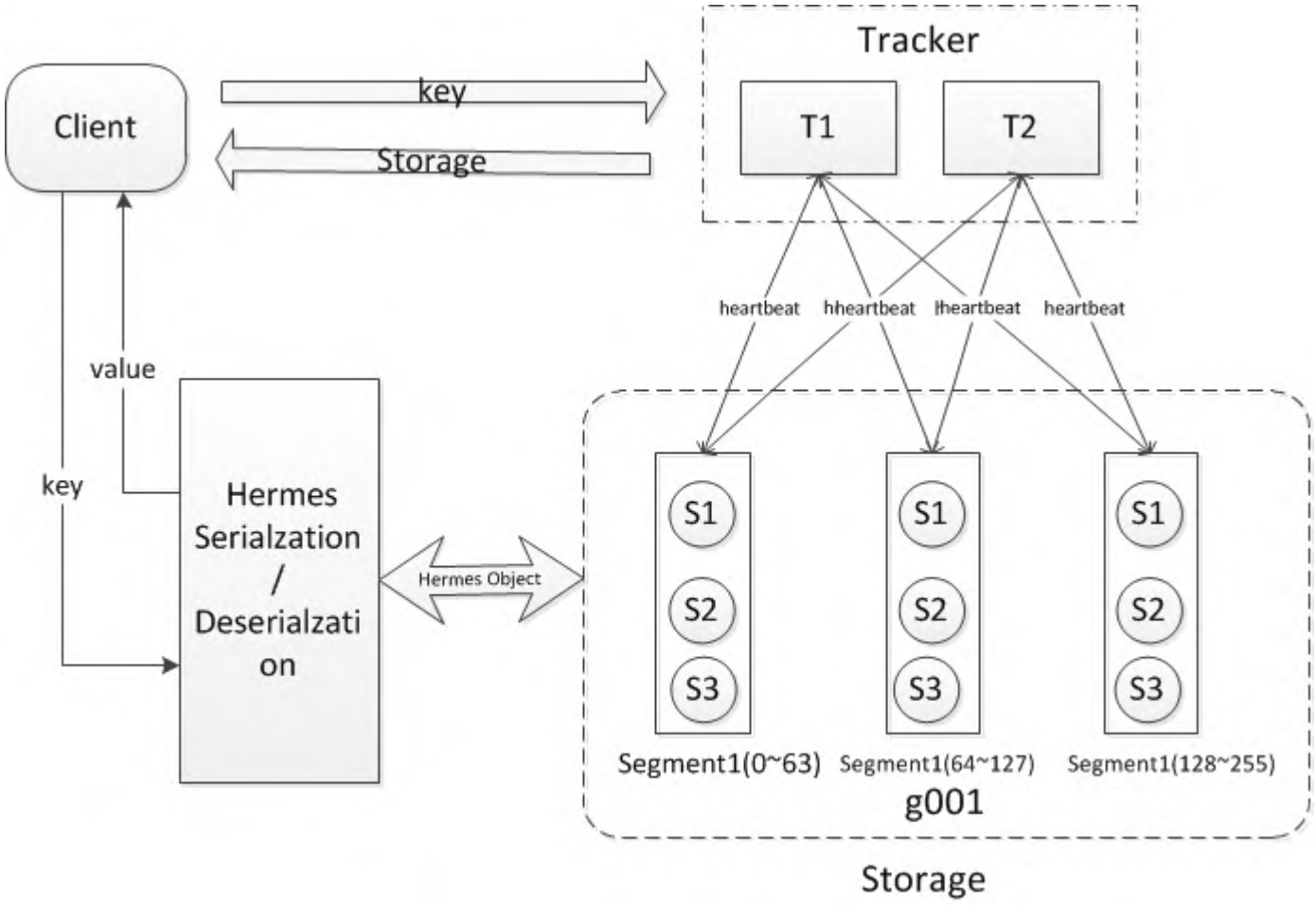
设计背景

- 以Redis为中心的主动缓存（全量缓存），随着业务的增长需要不断增加大内存机器来支撑整个集群
- 业务端对缓存的实时性要求不是很苛刻，但是需要支撑高并发的数据访问（多读少写）

设计目标

- 以磁盘作为数据载体
- 支持高并发数据访问
- 支持系统宕机快速恢复
- 支持List和Map类型的数据结构的基本操作
- 支持数据容灾
- 支撑负载均衡

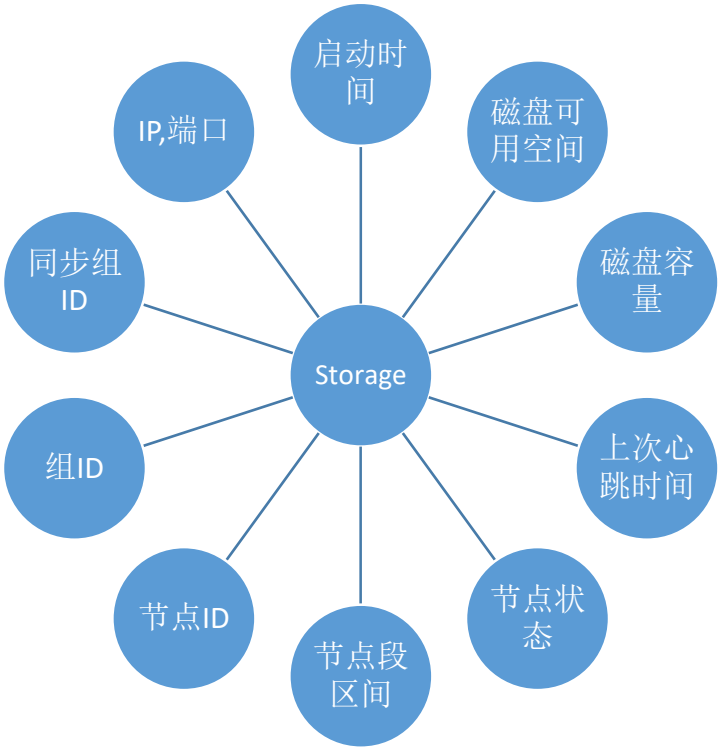
总体架构


Tracker
T1
T2
Client
key
Storage
heartbeat
hheartbeat
lheartbeat
heartbeat
value
key
Hermes
Serialzation
/
Deserialzati
on
Hermes Object
S1
S2
S3
S1
S2
S3
S1
S2
S3
Segment1(0~63)
Segment1(64~127)
Segment1(128~255)
g001
Storage

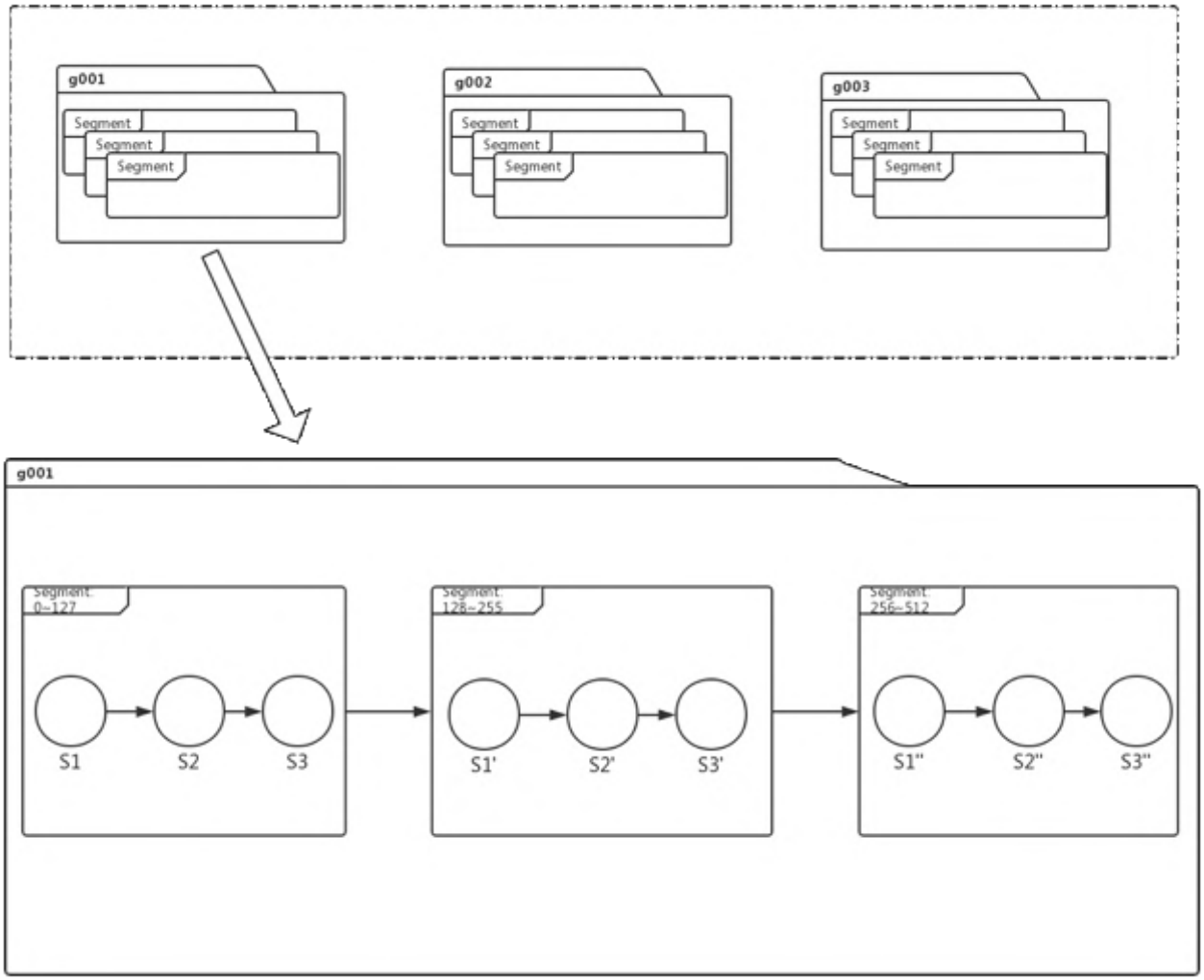
Tracker服务

Tracker状态维护

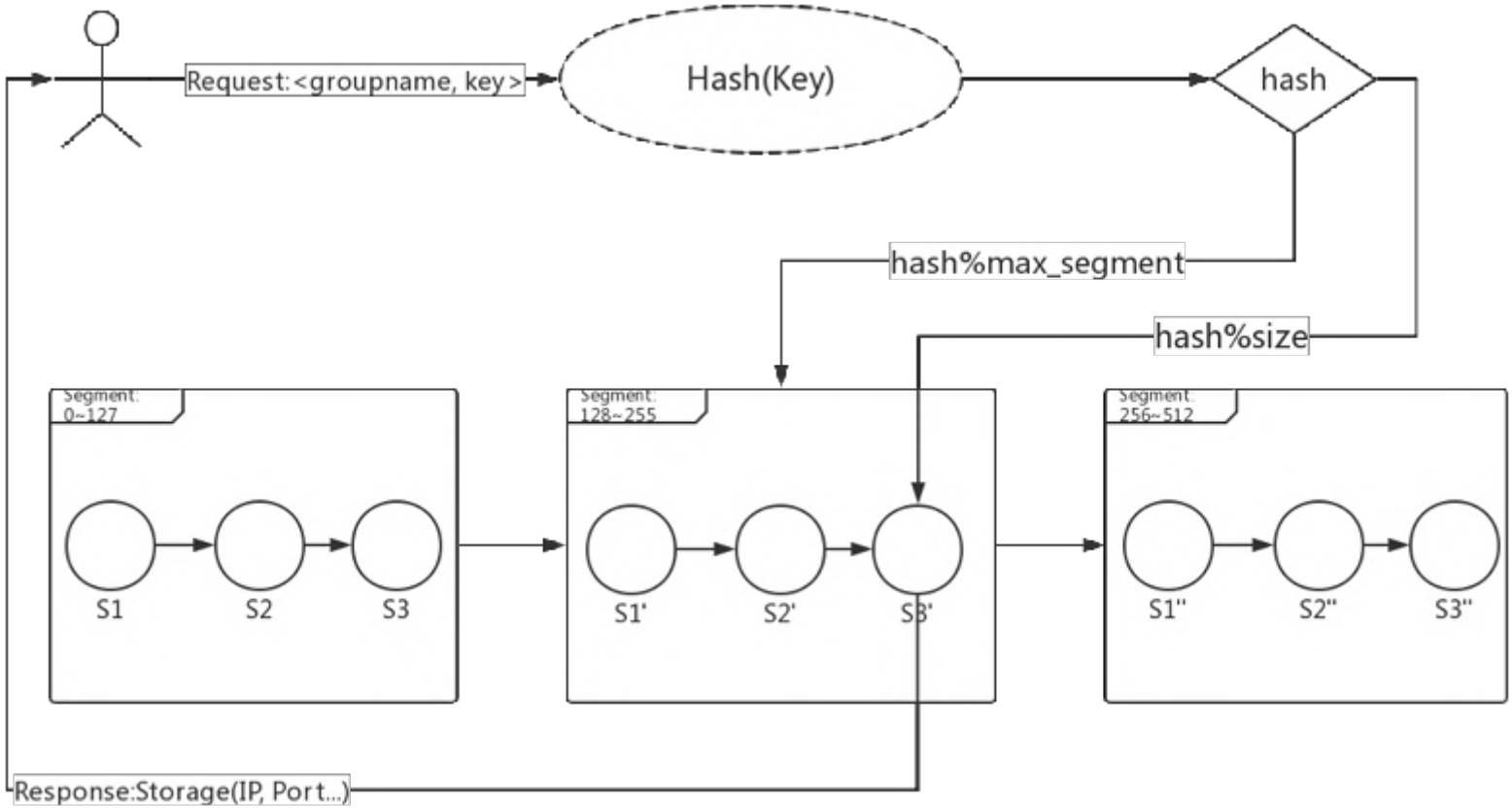
- 状态维护
- 负载均衡
- 同步支持
  - 同步组查询
  - 同步时间查询



Tracker状态数据结构



二次哈希负载均衡策略

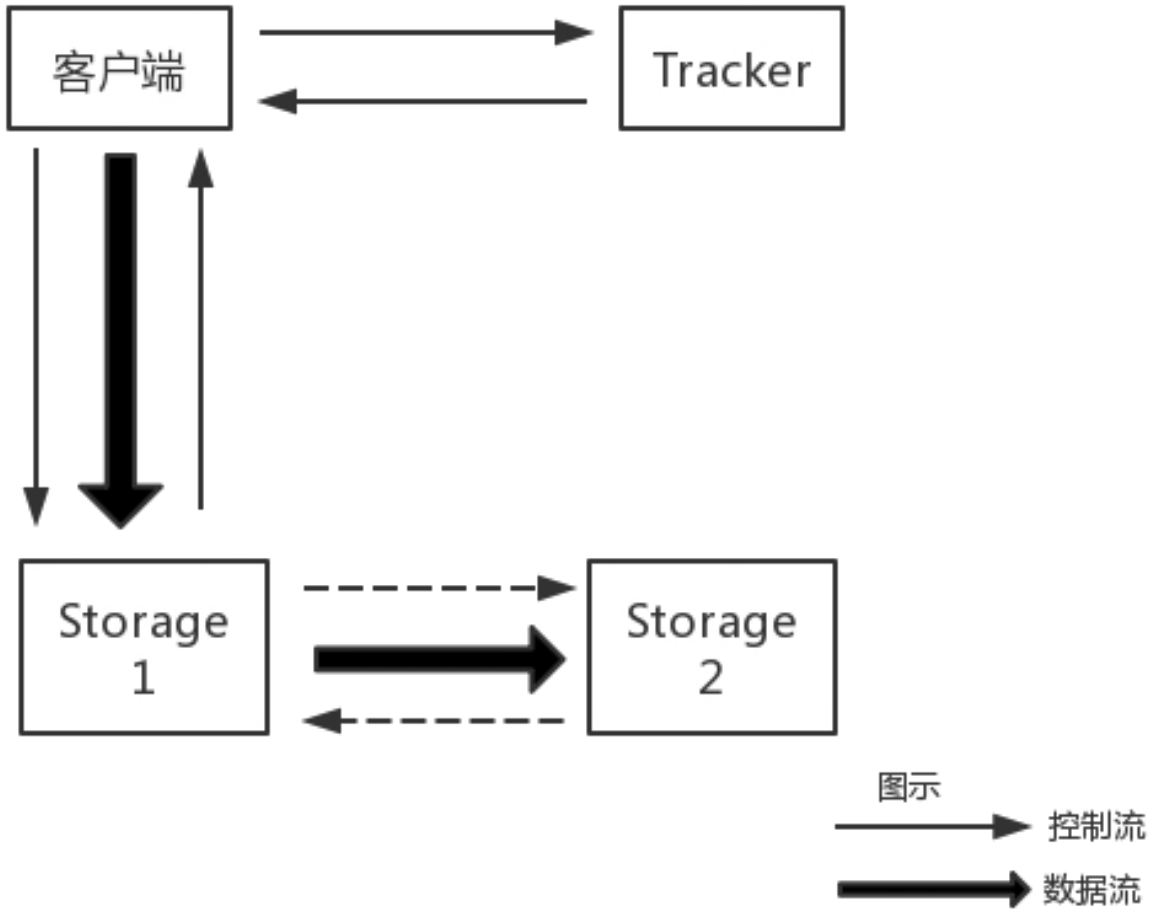

Request:<groupname, key>
Hash(Key)
hash
hash%max_segment
hash%size
Segment:
0~127
Segment:
128~255
Segment:
256~512
S1
S2
S3
S1'
S2'
S3'
S1''
S2''
S3''
Response:Storage(IP, Port...)

SDCC 2017深圳站
互联网应用架构实战峰会

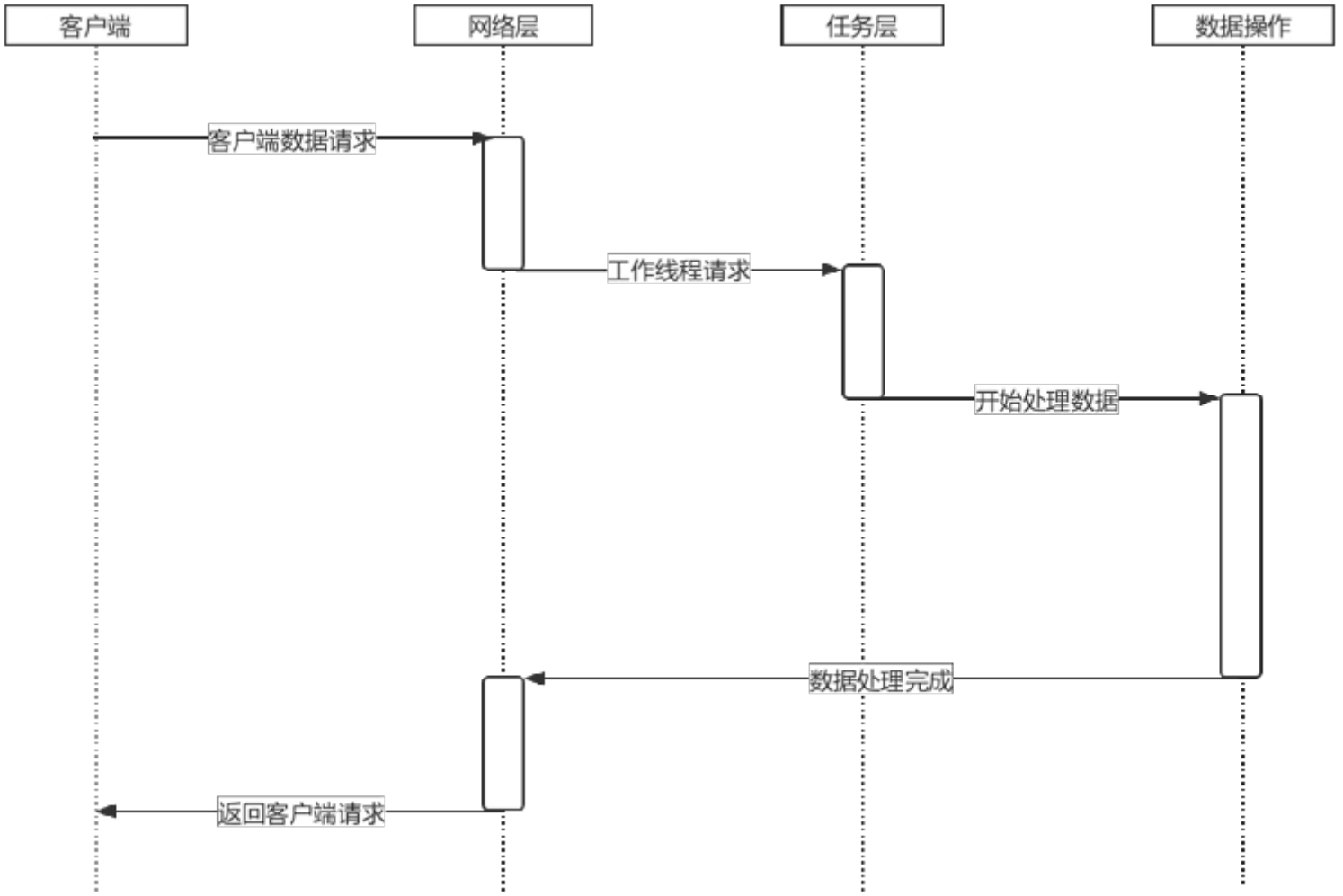
CSDN

数据存储

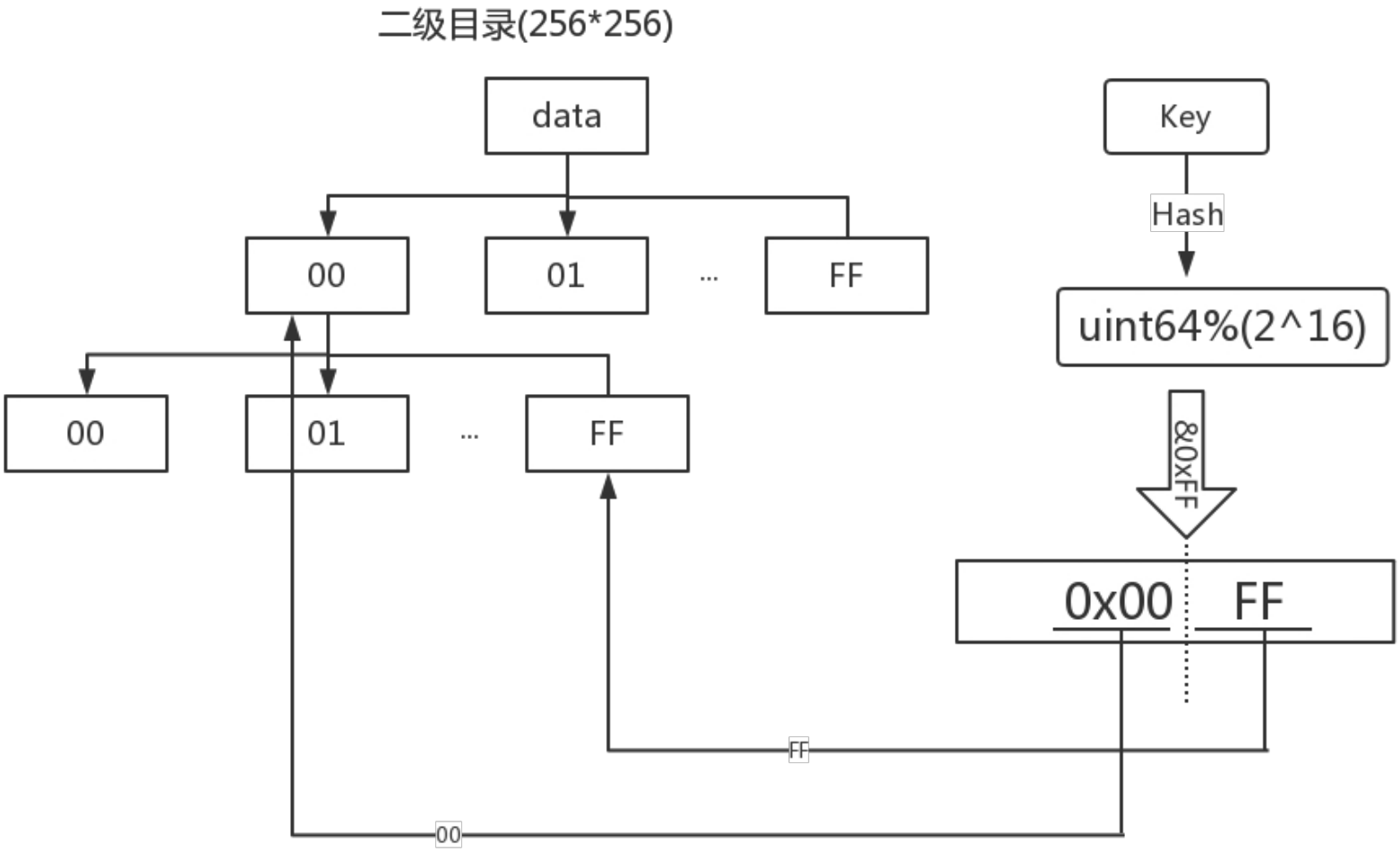
Lest数据和控制流



Storage数据服务端时序图



Storage的数据路由

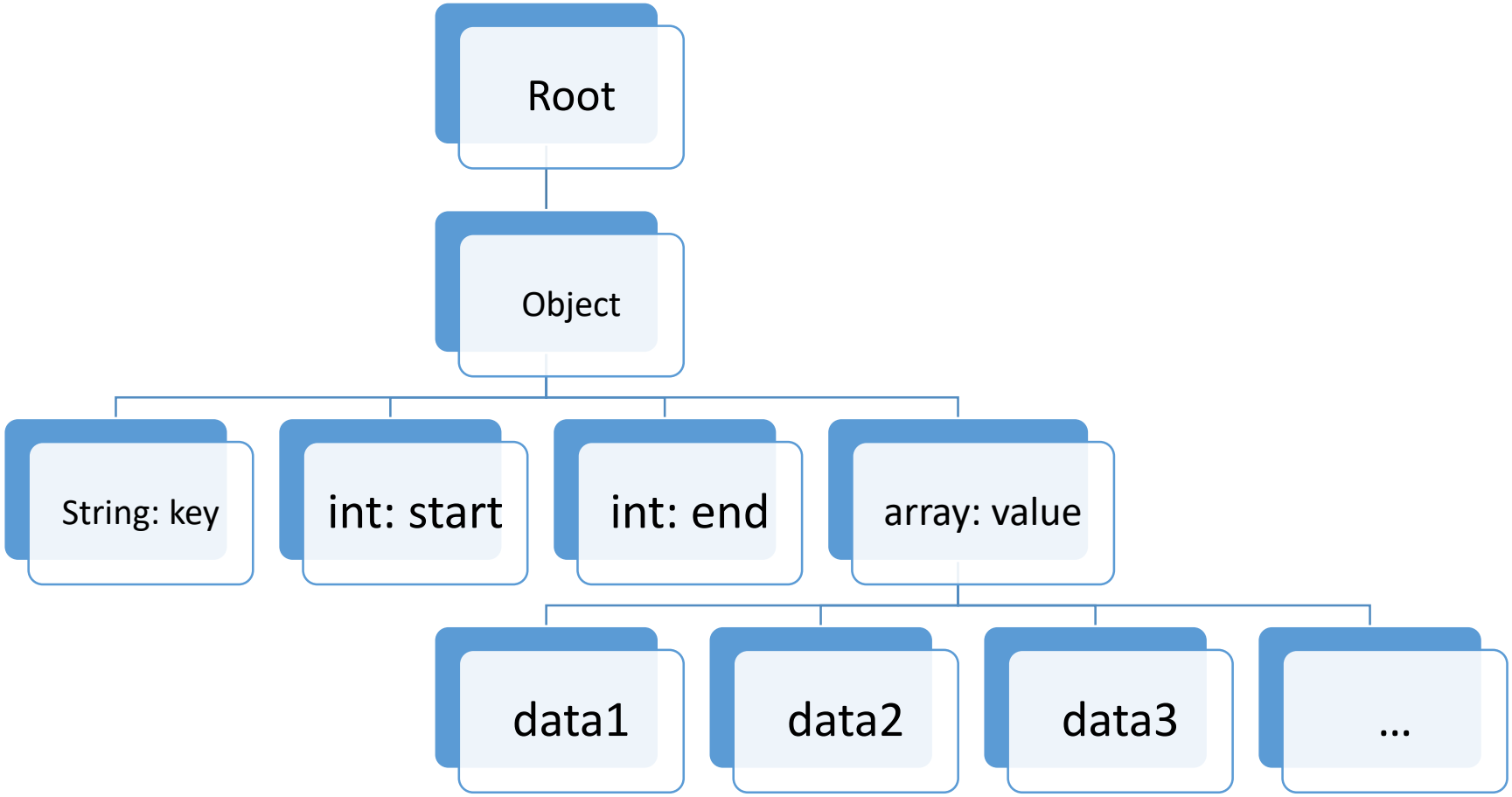


java客户端服务类

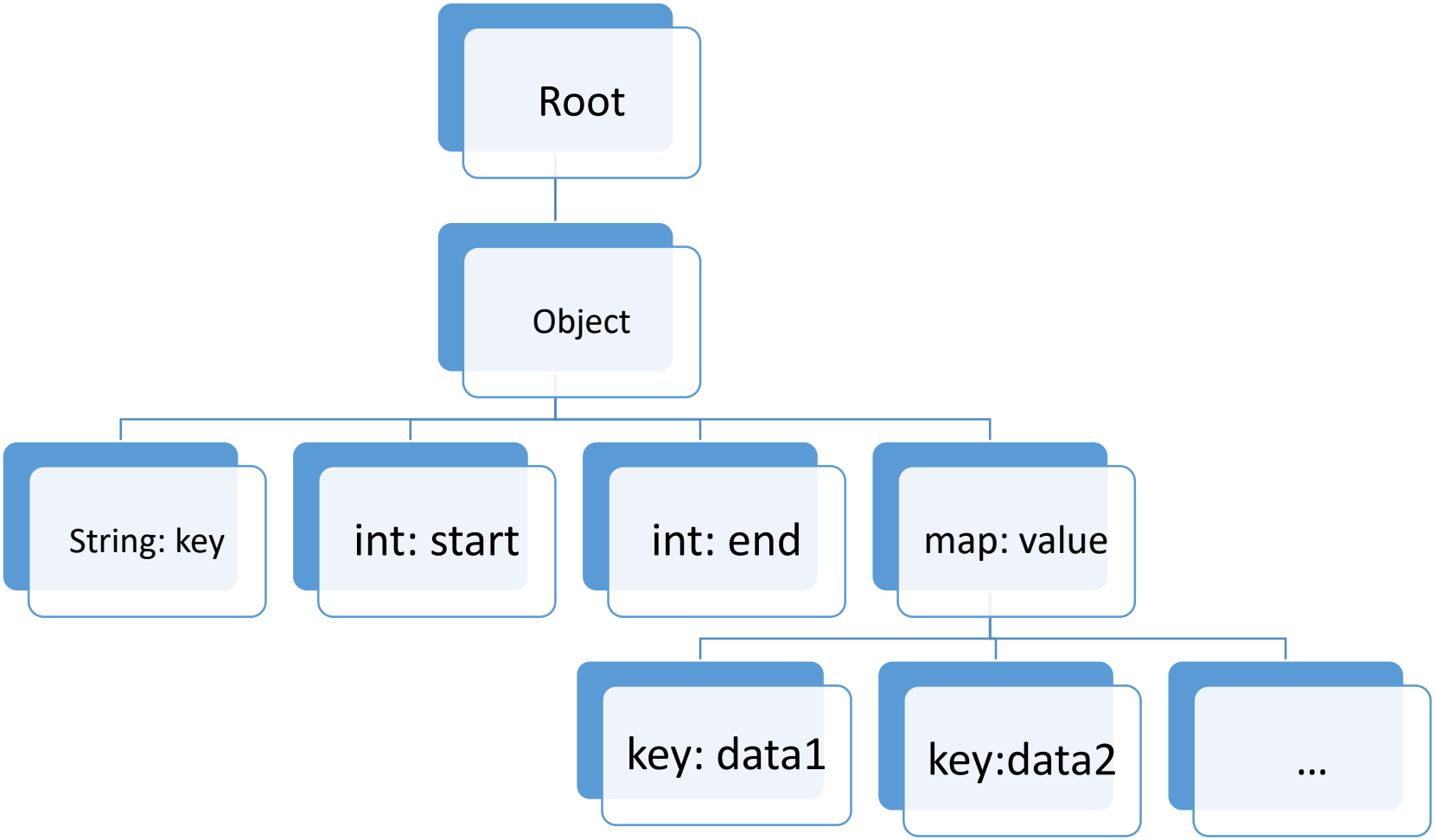
- Java LestList
  - Class LestListObject<T> {
    String key;
    int start;
    int end;
    ArrayList<T> value = new ArrayList<T>();
    …
  - }

- Java LestMap
  - Class LestMapObject<T> {
    String key;
    int start;
    int end;
    Map<K, T> value = new HashMap<K, T>();
    …
  - }

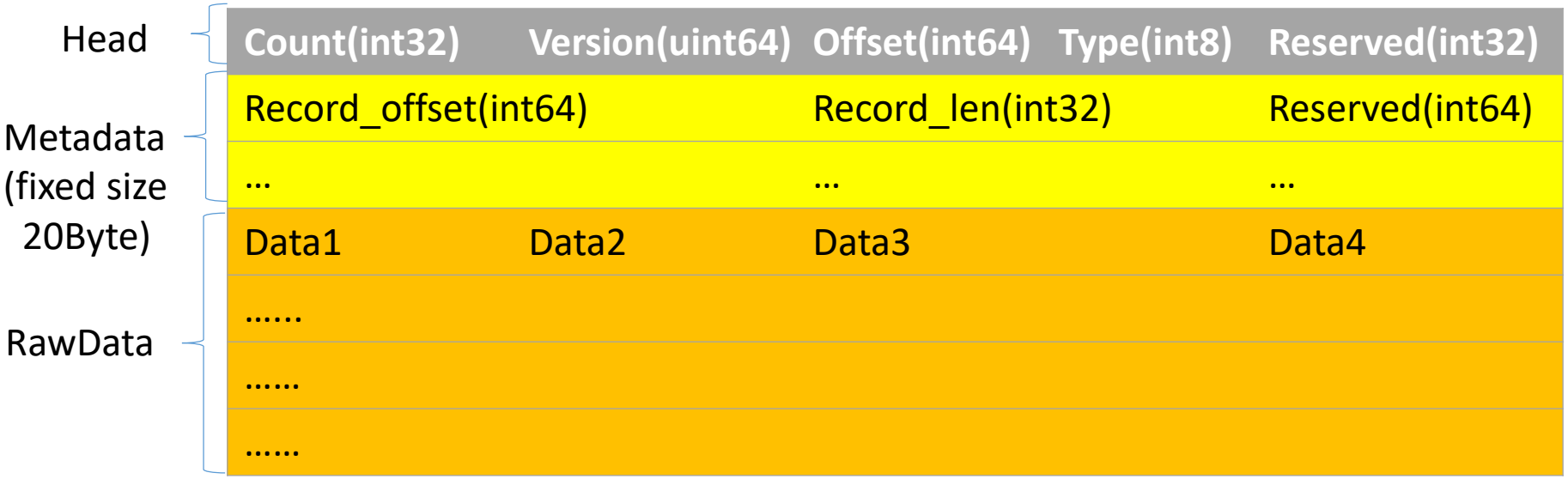
Storage服务端对象树——List



Storage服务端对象树——Map



List数据存储格式


Head
Metadata (fixed size 20Byte)
RawData
Count(int32)
Version(uint64)
Offset(int64)
Type(int8)
Reserved(int32)
Record_offset(int64)
Record_len(int32)
Reserved(int64)
…
…
…
Data1
Data2
Data3
Data4
……
……
……

Map数据存储格式

Head	Count(int32)	Version(uint64)	Offset(int64)	Type(int8)	Reserved(int32)
Metadata (fixed size 36Byte)	ID(int64)	Record_offset(int64)	Record_len(int32)	Version(uint64)	Reserved(int64)
	.....	.....	.....	.....	.....
RawData	Data1	Data2	Data3		Data4
	.....				
	.....				
	.....				

Storage的数据服务类型List

操作类型	参数	返回状态	方法说明	支持类型
Put	Key， List	成功或失败	根据Key存储List	List
Get	Key	List对象或失败	根据Key获取List	List
GetRange	Key， Start，End	[start, end)范围内的List对象或失败	根据Key获取[start, end)范围内的数据List	List

Storage的数据服务类型Map

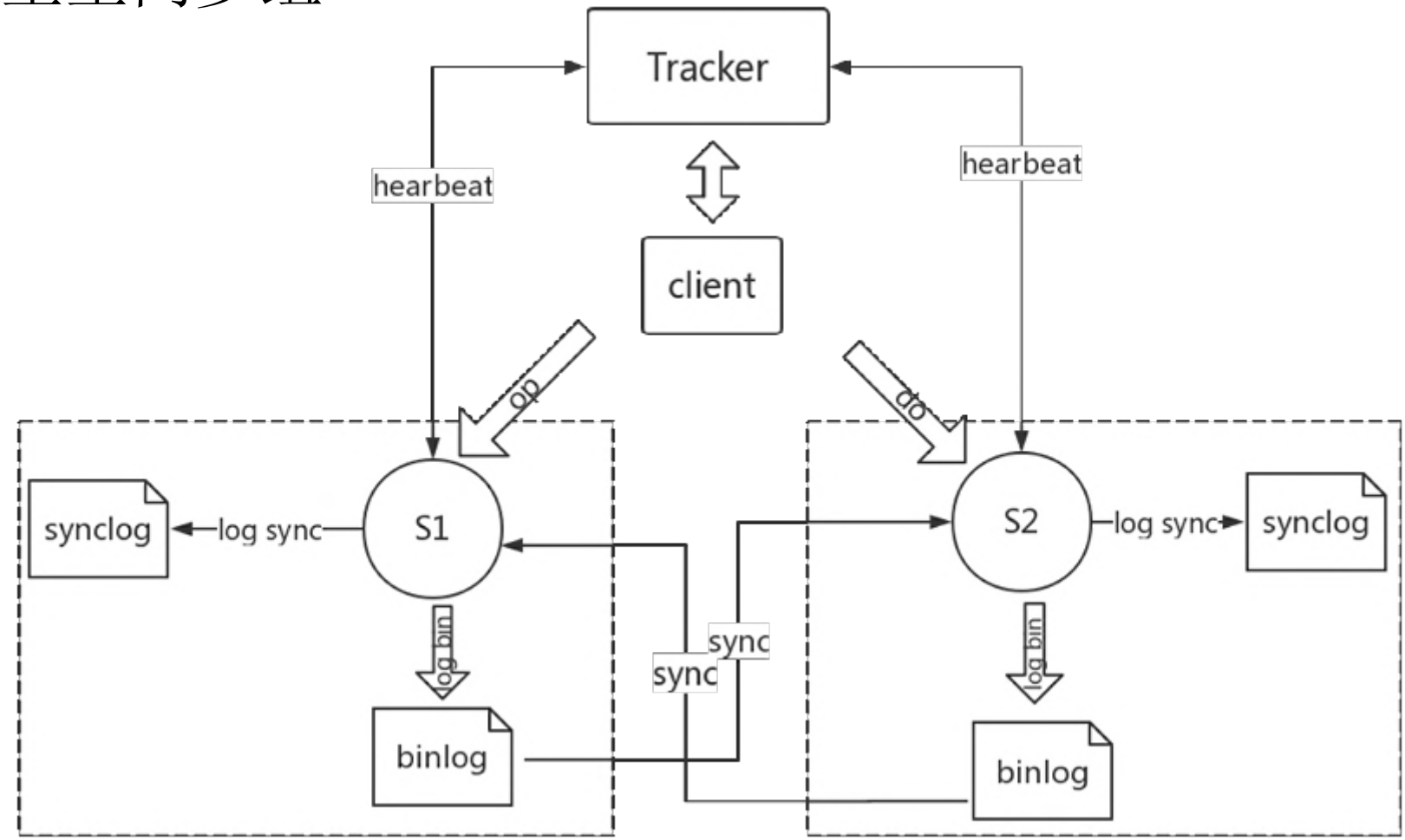
操作类型	参数	返回状态	方法说明	支持类型
MapPut	Key，Map	成功或失败	根据Key存储Map	Map
MapGet	Key	Map对象或失败	根据Key获取Map	Map
MapUpdate	Key，Map	成功或失败	根据Key更新相应的Map	Map
MapPutOrUpdate	Key，Map	成功或失败	根据Key存储或更新Map	Map
MapGetById	Key，ID	指定对象或失败	根据Key和ID获取该Map中的某一item	Map

数据同步

同步方式

- 状态转移（控制简单，传输数据量大）
  - 1 + 1 = 2 => 2
  - 2 + 1 = 3 => 3
- 复制状态机（控制复杂，传输数据小）
  - 1 + 1 = 2 => +1
  - 2 + 1 = 3 => +1

主主同步组


Tracker
hearbeat
hearbeat
client
op
op
synclog
log sync
S1
S2
log sync
synclog
log bin
log bin
sync
sync
binlog
binlog

Binlog/Synlog文件样式

操作	文件类型	Key	ID
ADD	L	Hello_List	
ADD	M	Hello_Map	
UPDATE	M	Hello_Map	0
UPDATE	M	Hello_Map	1
UPDATE	M	Hello_Map	2

SDCC 2017深圳站
互联网应用架构实战峰会

CSDN

性能测试

Lest性能测试

操作	平均长度/10k	平均耗时/10k	平均大小/10k	内存占用
Put	1007	84ms	245KB	20M
Get	1007	8ms	245KB	20M
MapPut	995	38ms	242KB	20M
MapGet	995	8ms	242KB	20M
MapGetByID	1	1ms	249Byte	20M

Lest优势

- 占用内存小
- 能够快速恢复，不用重构内存数据结构
- 同步组采用主主模式，有利于负载均衡
- 使用更加轻量级的二进制序列化，有利于数据的存储和传输
- 在服务器端构建了对象树，方便数据操作

有待改进

- 随机读写的问题，后面考虑顺序写，读缓存
- 同步组中的不能动态添加storage节点，后面考虑强一致性同步
- 数据同步过程中批量更新，目前采用单条同步，后面考虑改成批量同步

Q&A