简单得不像技术活

--风险检测中时间窗口计算

Geekbang.
极客邦科技

整合全球最优质学习资源，帮助技术人和企业成长

Growing Technicians, Growing Companies

InfoQ

专注中高端技术人员的
技术媒体

EGO EXTRA GEEKS' ORGANIZATION NETWORKS

高端技术人员
学习型社交网络

StuQ

实践驱动的
IT职业学习和服务平台

GiT GEEKBANG INTERNATIONAL TRAINING 极客邦培训

一线专家驱动的
企业培训服务

地址：北京市朝阳区洛娃大厦C座6层1607室　　网址：www.geekbang.org

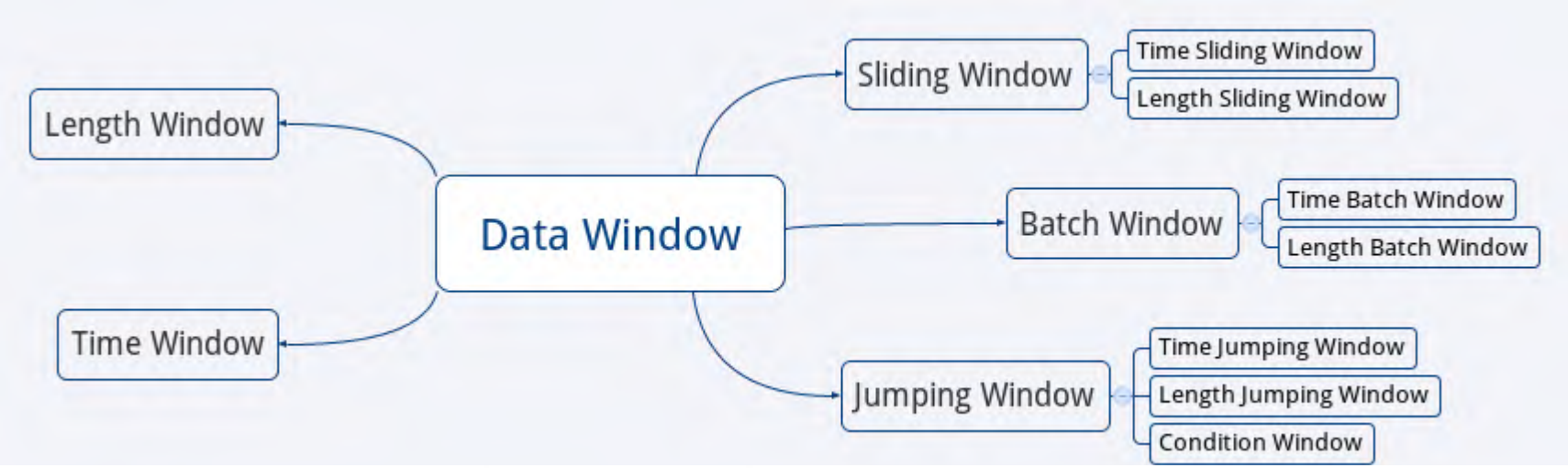
刷脸

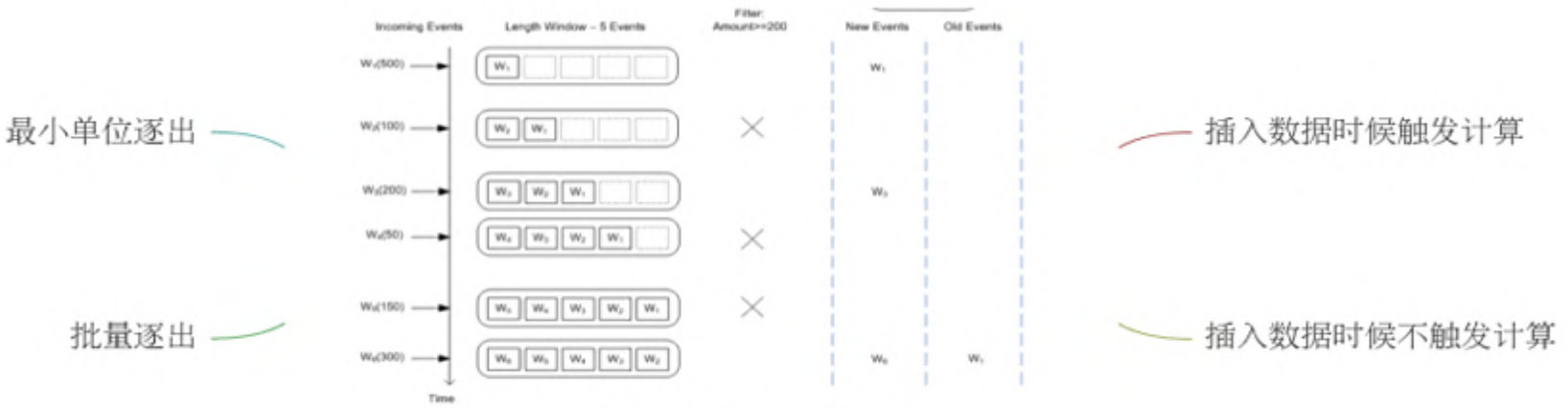
- 安全圈的新人
- 业务风控后端

Agenda

- 风控侧写
- 时间窗口统计的需求场景和各种实现
  - 点击欺诈和布隆过滤的一种改进
  - 支付欺诈和区间树的应用技巧
  - 区间树在内存使用上的优化
  - 通用场景下灵活性和实时性兼顾的方案
  - 818黄牛抢购的风险特征（值）计算的案例数据
- 一个通用的内存分布式框架介绍

窗口数据计算





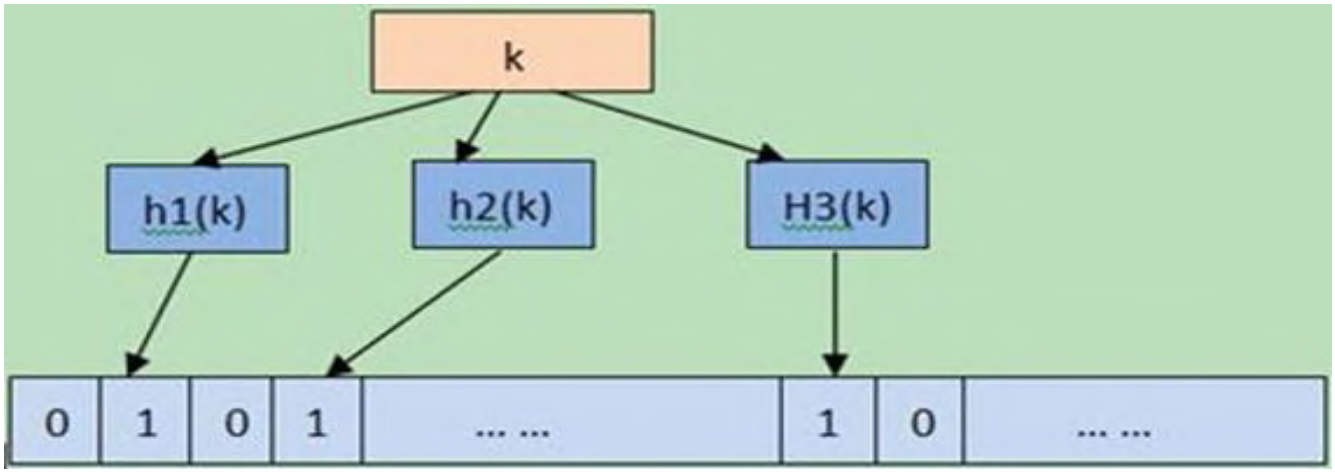
窗口数据计算模式

Case 1 点击欺诈中布隆过滤改良应用

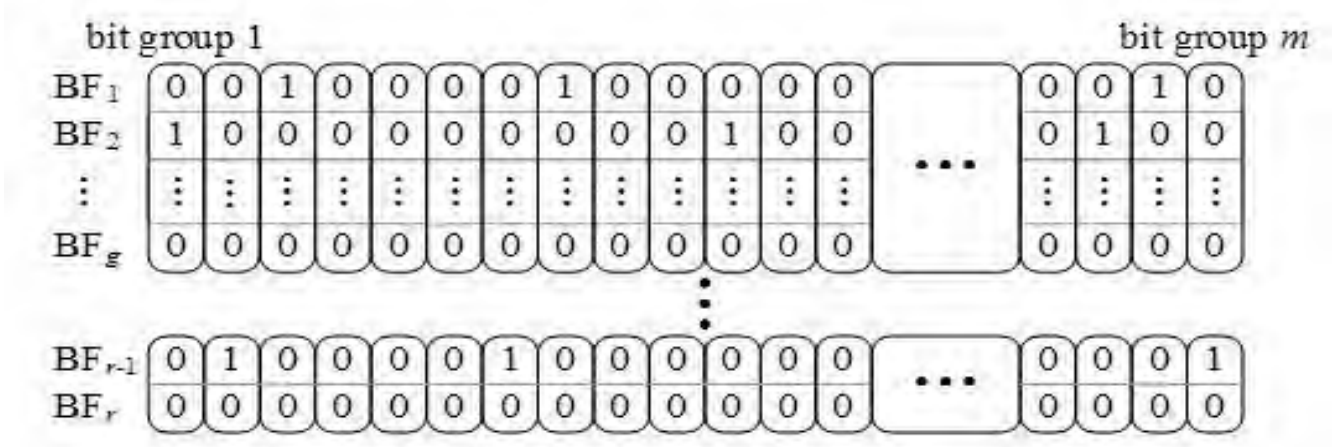
- 什么是点击欺诈
- 检测什么
  - 点击者身份
  - 重复点击
- 布隆过滤怎么用
  - ~~Daily~~

- 有限内存
- 任意时长（窗口长度）

Case 1 点击欺诈中布隆过滤改良应用 Cont.

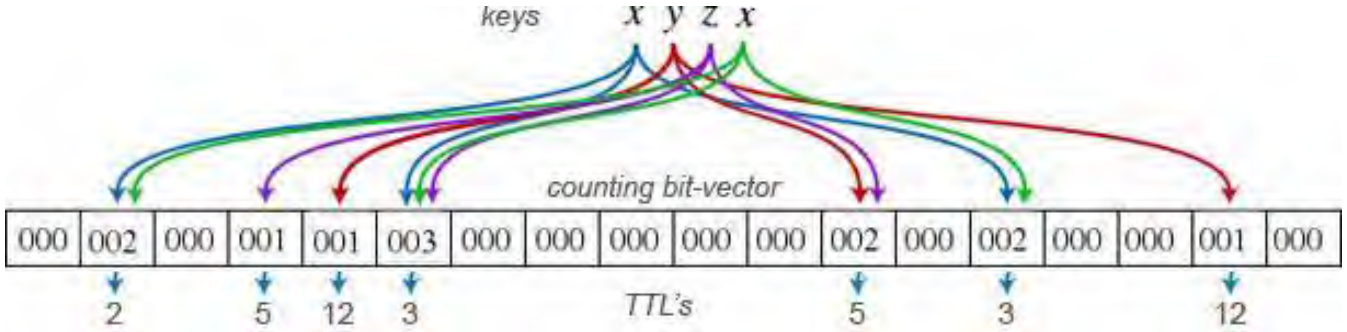


Hash 一次　（GBF）　读写 一次



Case 1 点击欺诈中布隆过滤改良应用 Cont.

- Timing Bloom Filte(r)



  - logN + C
  - 32位的counter足够记录130年
  - 2 ×（logN + 1）

布隆过滤的其他几种改进版本

- CBF
- SBF
- DCF

是 数据结构

更是 算法

Case 2 支付欺诈和区间树的应用技巧

- 判断支付欺诈的一个有效特征
  - 时间序列上额度变化分析

$\Delta\delta$

求导

时间 time	用户 ID 号 (user_id)	交易金额 bill_amount	ip 地址
2011-1-21 19:50	111111111	1200	192.168.1.10
2011-11-21 19:35	111111112	1500	192.168.1.10
2011-11-21 20:01	111111113	10000	192.168.1.11
2011-11-21 20:00	111111111	4215	192.168.1.10
2011-11-21 20:01	111111115	2310	192.168.1.13
2011-11-21 20:01	111111116	600	192.168.1.10
2011-11-21 19:58	111111112	2651	192.168.1.15
2011-11-21 20:01	111111118	1921	192.168.1.16
2011-11-21 19:50	111111119	6354	192.168.1.17
2011-11-21 20:03	111111111	4902	192.168.1.12

Case 2 支付欺诈和区间树的应用技巧

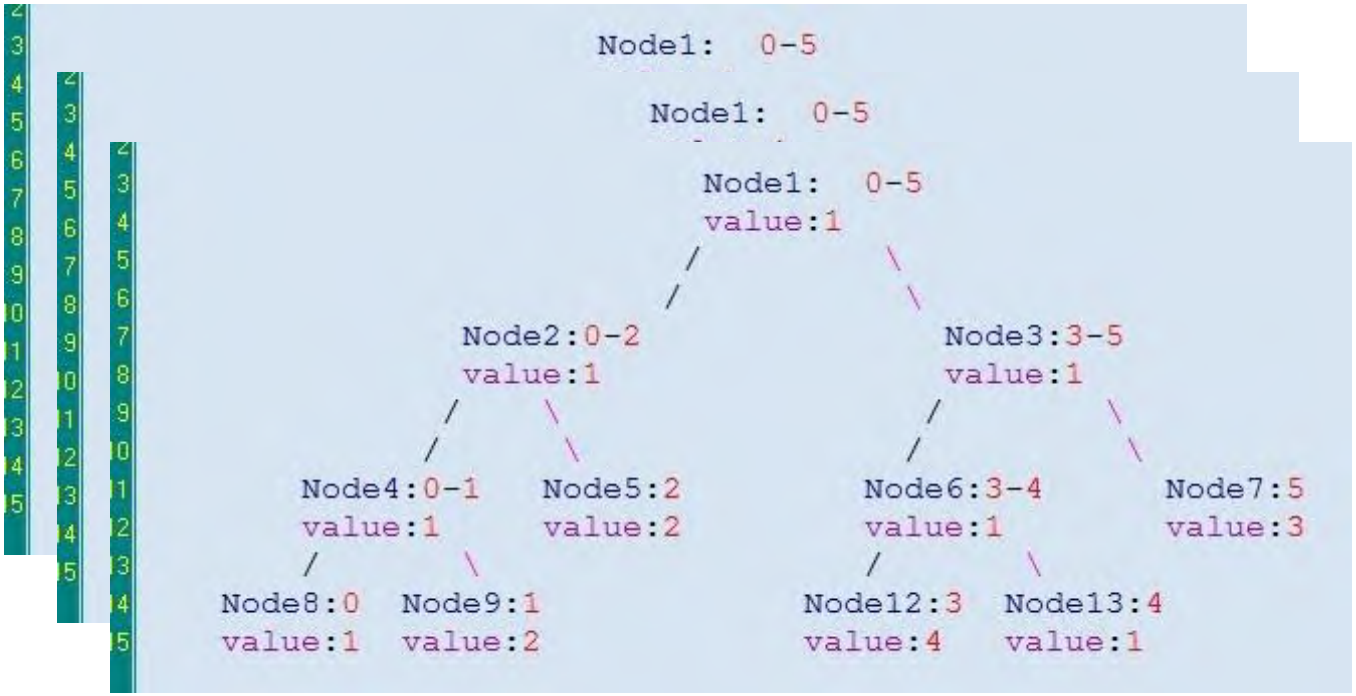
Cont.

（key，value，Cal Funcs）

（用户id：11111118，交易金额，Sum）

- 普通算术函数
  - sum
- 比较函数
  - max
- 统计函数
  - count

Distinct



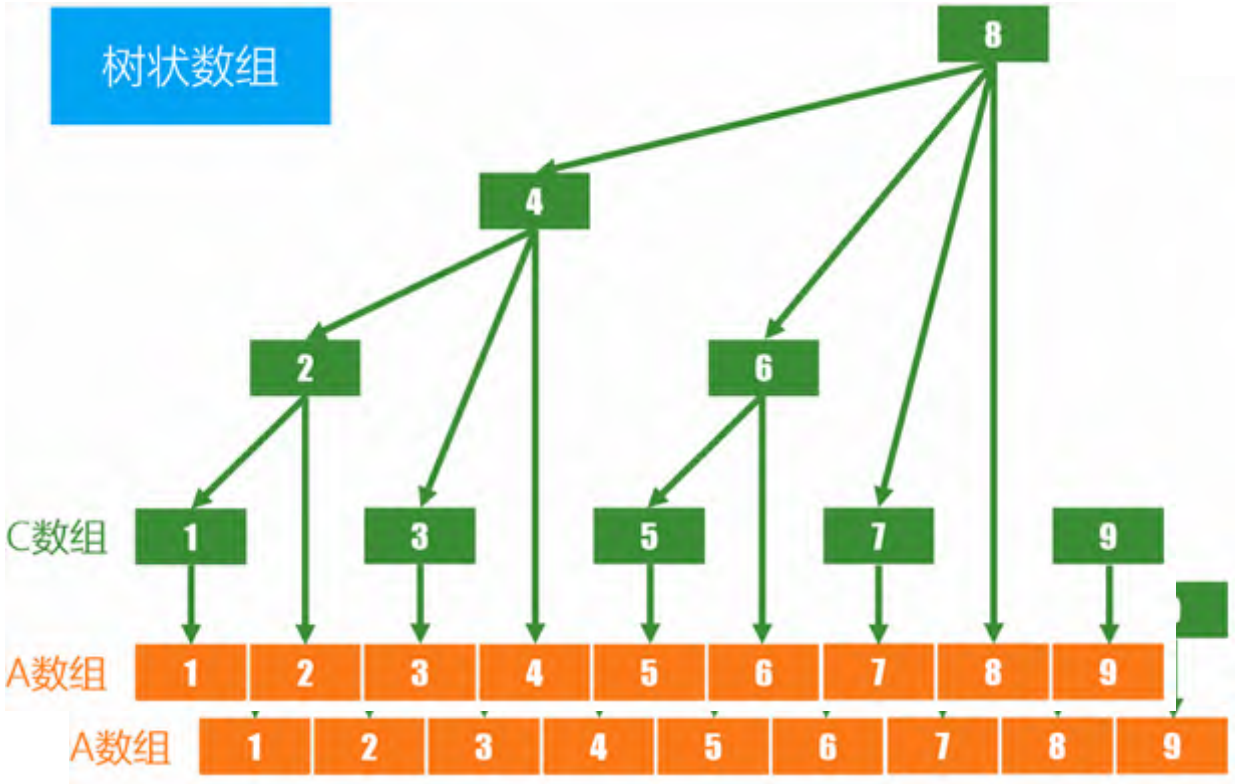
每个叶子节点：最小时间单位（精度）

Case 2 支付欺诈和区间树的应用技巧 Cont.

- 10年末
- 单机250G
- 3台物理服务器
- 计算延迟<10ms
- 4G的虚机=运维麻烦

Case 3 区间树在内存使用上的优化

- 内存占用更少
- 数据结构更简单
- 计算函数可累积
- 适用场景受限



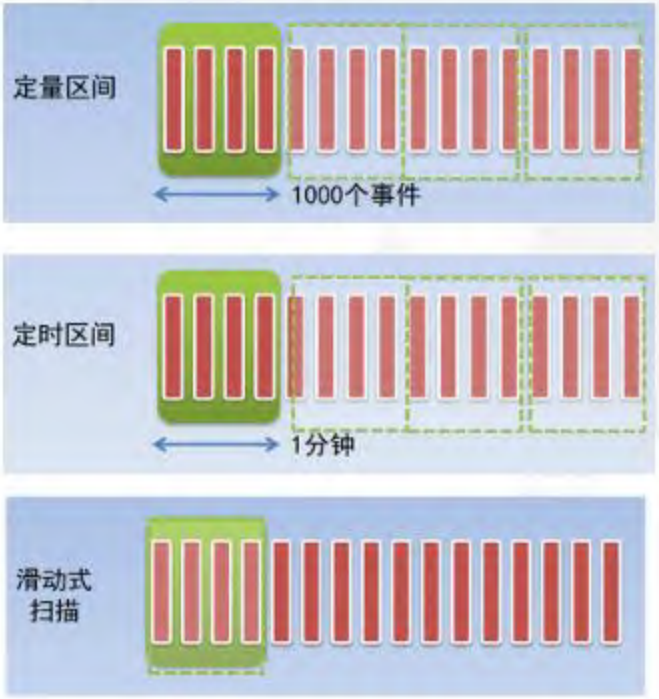
时间长度上通过2倍空间，避免移除操作

窗口运算的通用性抽象和流式计算:Libra

- 事件驱动的设计方法
- SEDA (staged event-driven architecture)
- 消息驱动的架构设计

实时计算平台Libra以及使用情况

```
select count(distinct client_id) as device_count, collect_time from
apm_http_data.win:time_batch(15 minute) group by instance_code
```



201504060801

1 00 1

818的实际数据

- 3亿次登陆×6个实时指标/天，4w TPS峰值，计算延迟95%低于10ms，99%低于30ms
- 百万下单×60+窗口指标/天
- 支付×20+窗口指标/天
- 800+计算节点

降维（打击？）

- 全息数据 ==> 特征数据
- Kernel函数
- 机器学习？

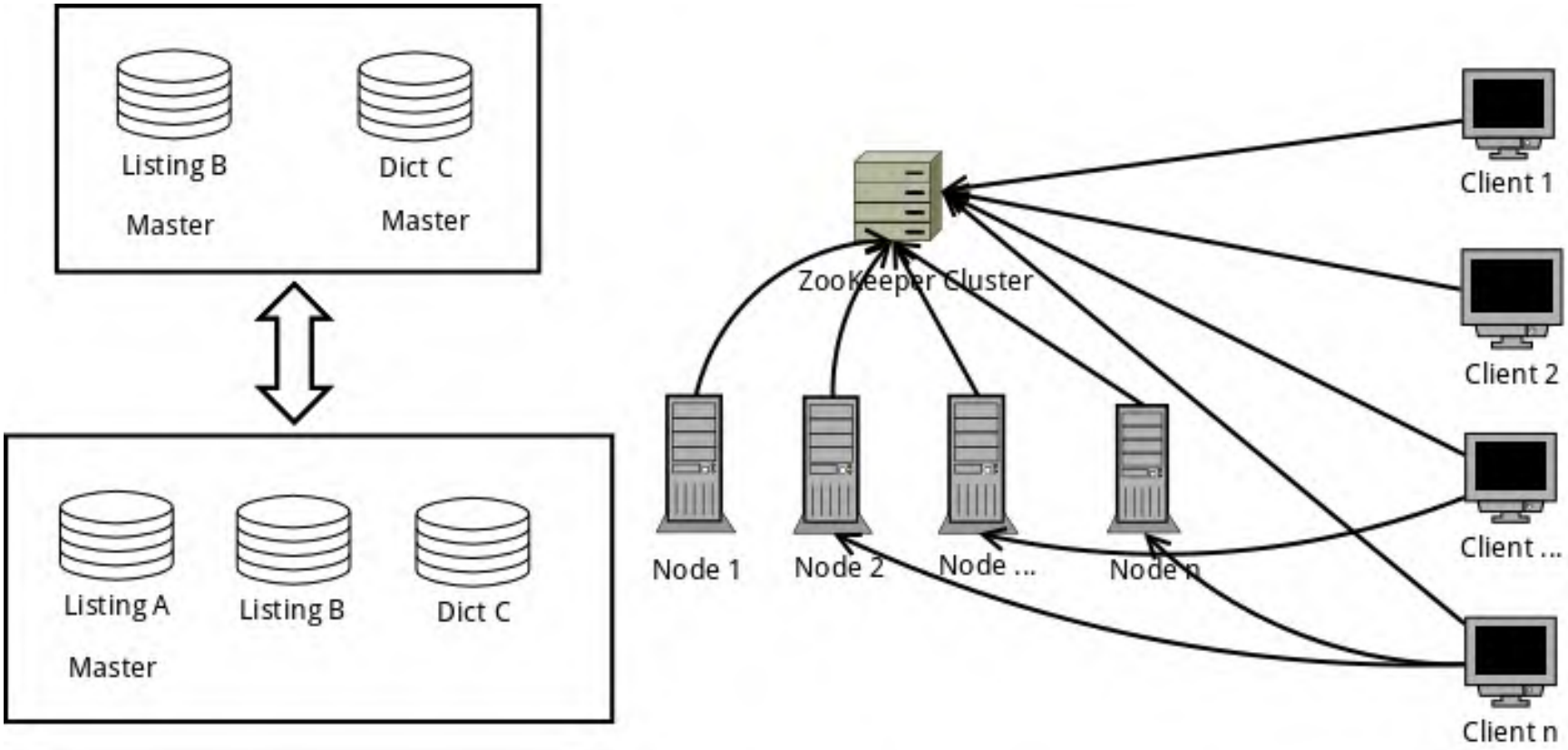
A Generic Dynamo Framework

- 统一的具备水平扩展的访问接口
- 抽象共通的操作接口
- 提供备份之间的最终一致性实现
- 自定义的初始化，持久化能力
- 自动增加新备份的能力？自定义？
- 支持的数据结构

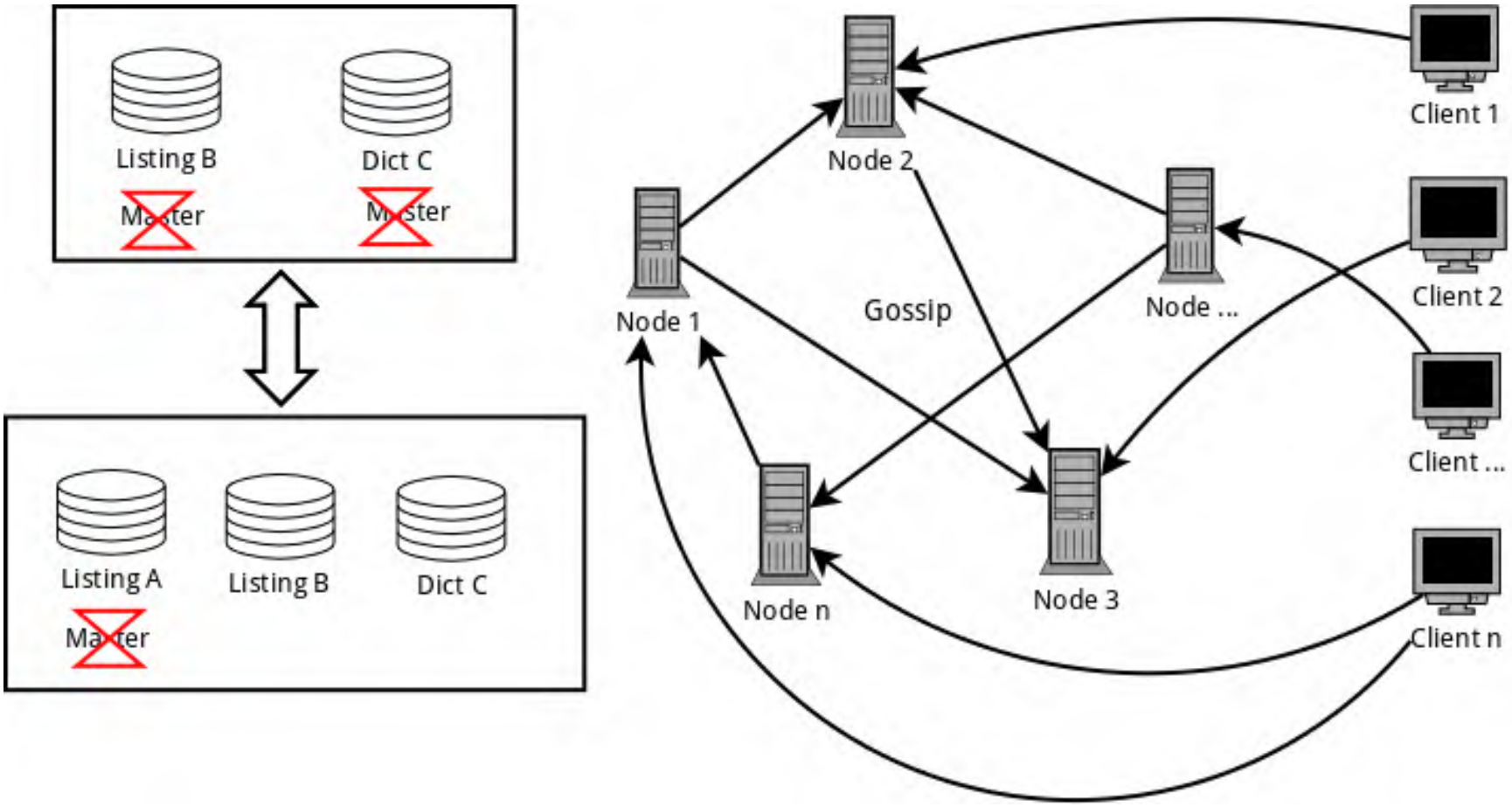
Specification 2 Outline of operation-based object specification. *Preconditions, return values and statements are optional.*

```
 1: payload Payload type; instantiated at all replicas
 2:     initial Initial value
 3: query Source-local operation (arguments) : returns
 4:     pre Precondition
 5:     let Execute at source, synchronously, no side effects
 6: update Global update (arguments) : returns
 7:     atSource (arguments) : returns
 8:         pre Precondition at source
 9:         let 1st phase: synchronous, at source, no side effects
10:     downstream (arguments passed downstream)
11:         pre Precondition against downstream state
12:         2nd phase, asynchronous, side-effects to downstream state
```

Approach A. base on zookeeper



Approach B. base on gossip and vector clock



Thanks!

附录

- http://www.eecs.harvard.edu/~mdw/papers/seda-sosp01.pdf
- http://hal.upmc.fr/docs/00/55/55/88/PDF/techreport.pdf
- https://github.com/oldratlee/translations/blob/master/log-what-every-software-engineer-should-know-about-real-time-datas-unifying/README.md
- http://wenku.baidu.com/link?url=FolbmG-0zvBmZivAy2XTAwLp15wJZW9RIVzNy4rJdCf4UpDJiXbAeKijNm0eurWQCkeZfqVJe5k5MZNzgxPlbN6PXdQkkw-jFvtm18Y6Kr_