GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

去哪儿网消息中间件演进

余昭辉
携程 去哪儿网

2017.thegiac.com

现状

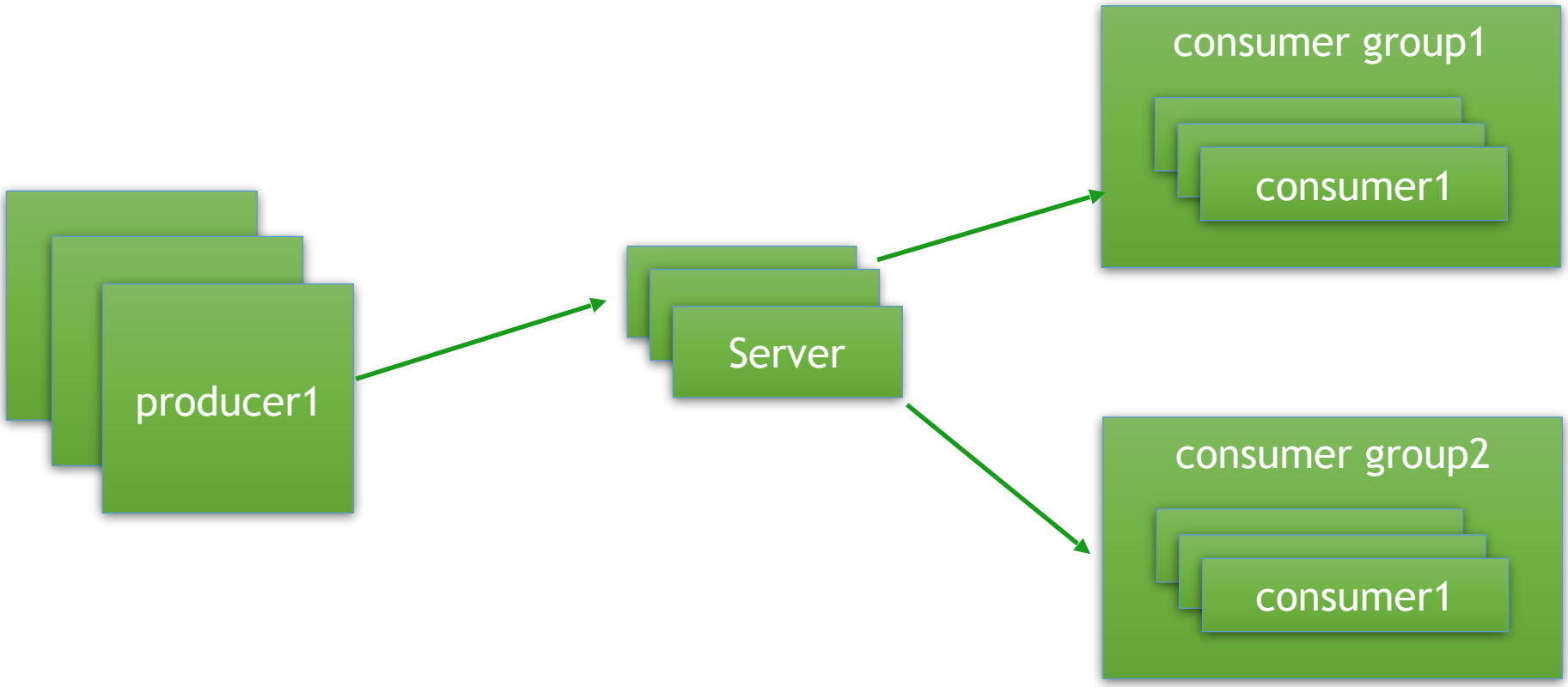
1W+ topic

35W+ QPS

280亿+ 每天

承载所有核心交易链路，完全消息驱动的交易系统

亦涉及搜索，报价等高流量业务

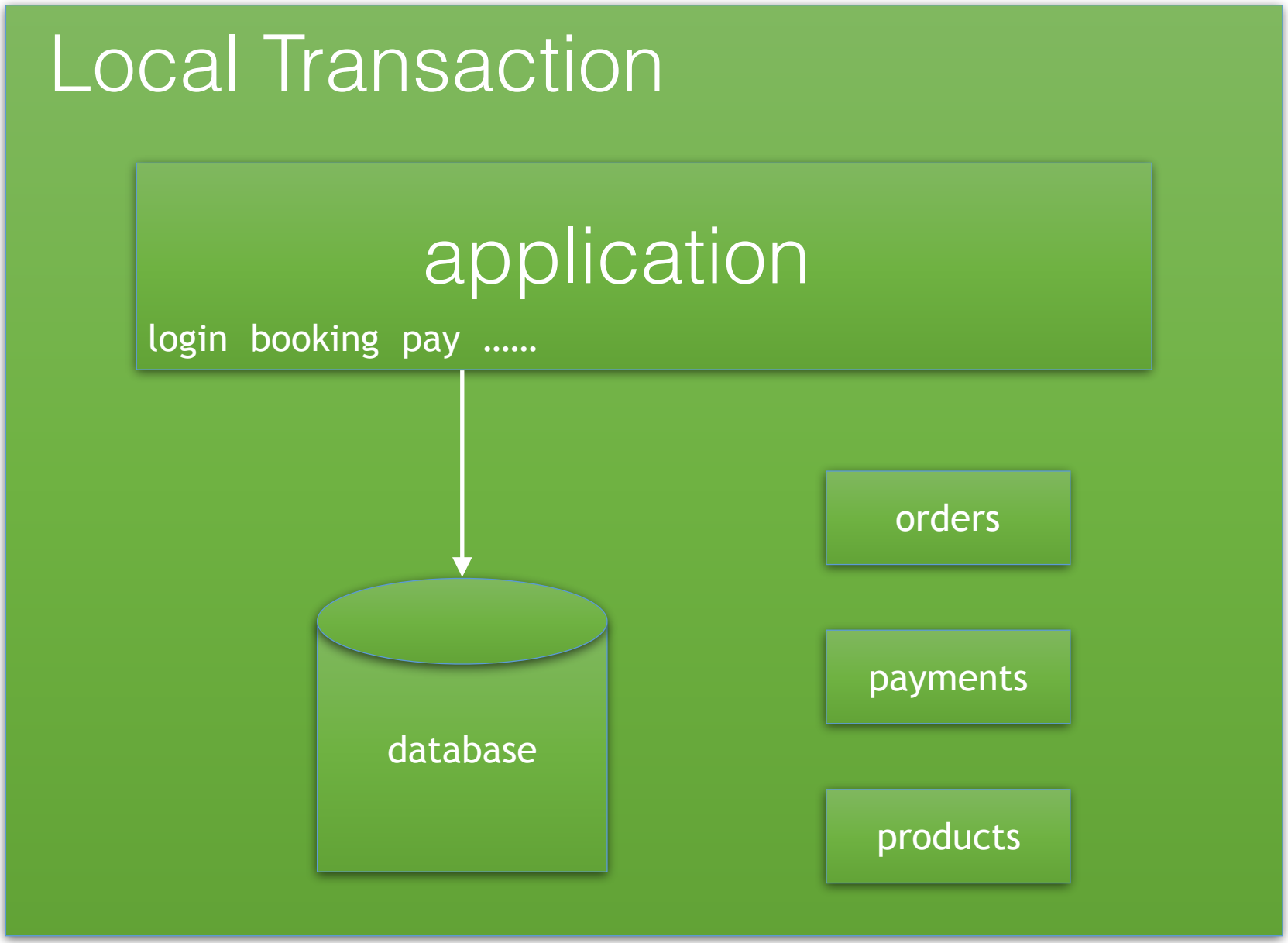

consumer group1
consumer1
producer1
Server
consumer group2
consumer1

大纲

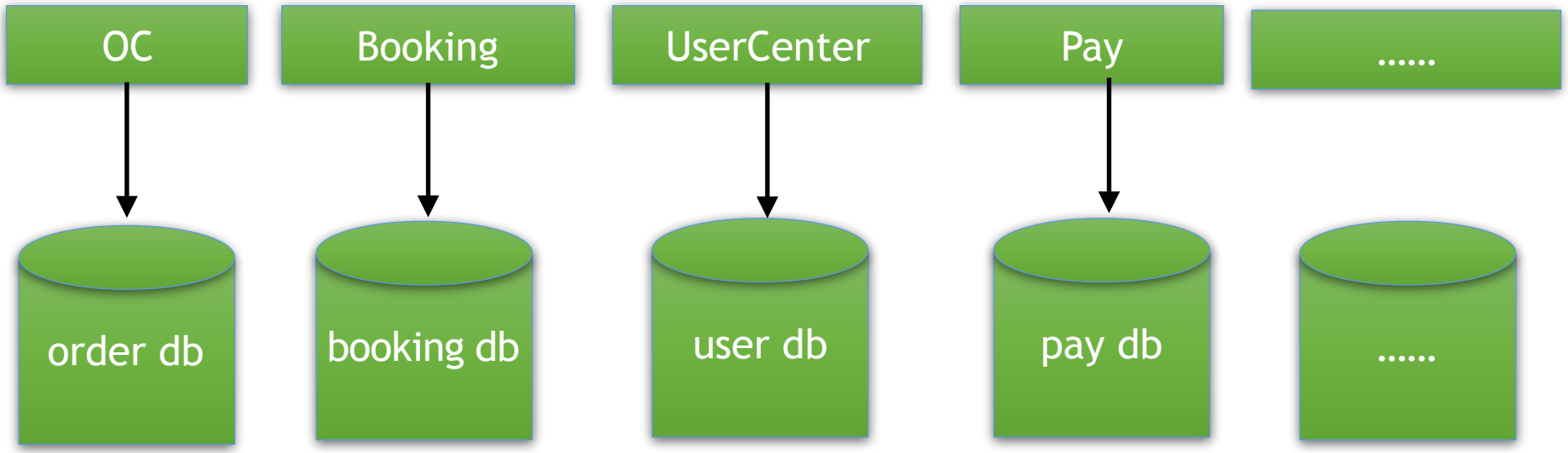
- 客户端设计
  - 一致性
- 存储设计
  - 横向扩展
  - 海量堆积

客户端设计 – 最终一致性

Local Transaction



```
@Trasnactional
public void pay(long orderId){
  payService.pay(orderId);
  orderService.updateStatus(orderId, PAY_SUCESS);
}
```


OC
Booking
UserCenter
Pay
.....
order db
booking db
user db
pay db
.....

```
@Trasnactional
public void pay(long orderId){
  payService.pay(orderId);
  orderService.updateStatus(orderId, PAY_SUCESS);
}
```

分布式事务?

BASE

—Ebay

E — 最终一致性

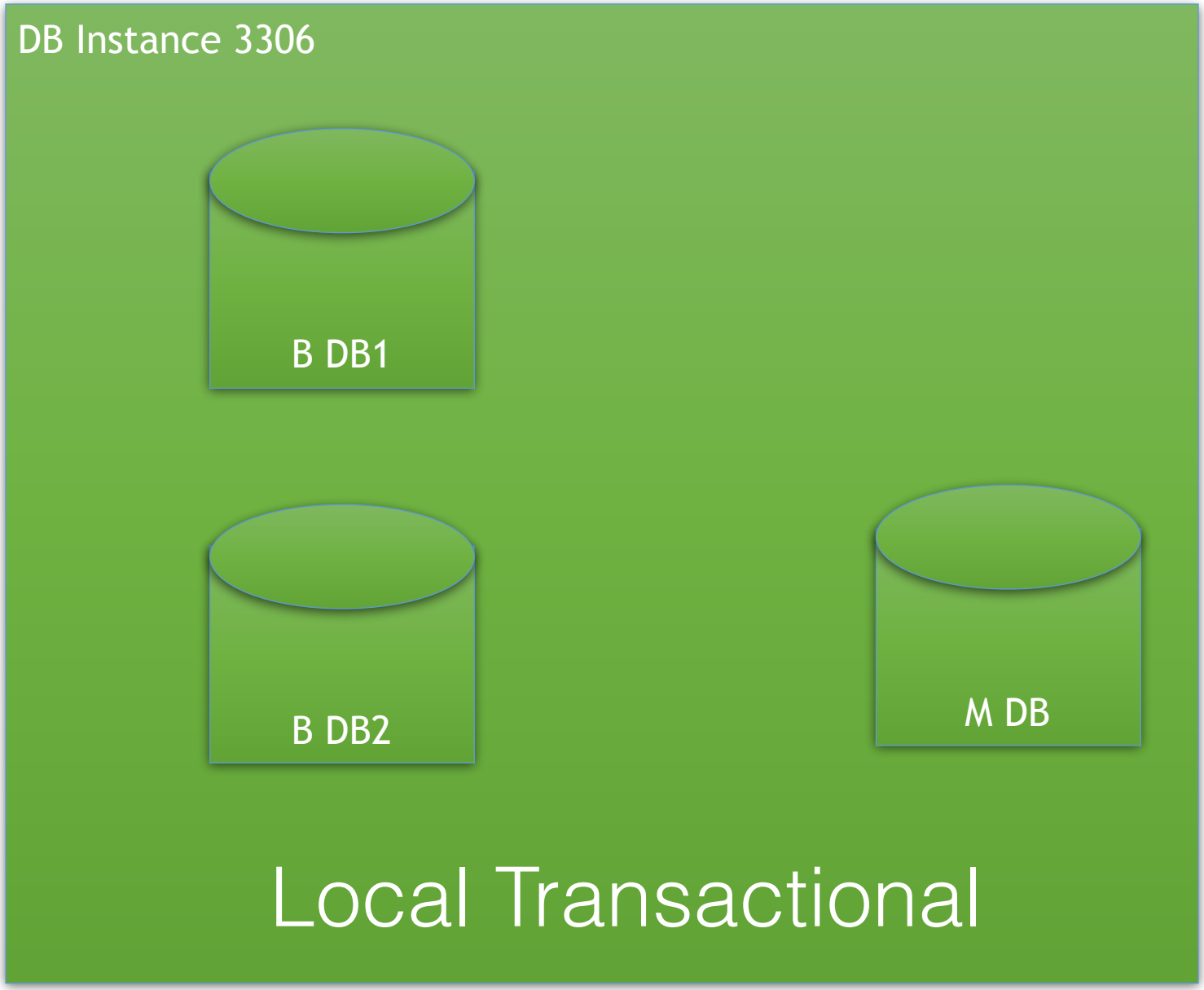
```
public void pay(long orderId){
  payService.pay(orderId);
  orderService.updateStatus(orderId, PAY_SUCCESS);
}
```

A Job
Step1. scan payments table
Step2. check order status
Step3. update lost status(retry to success)

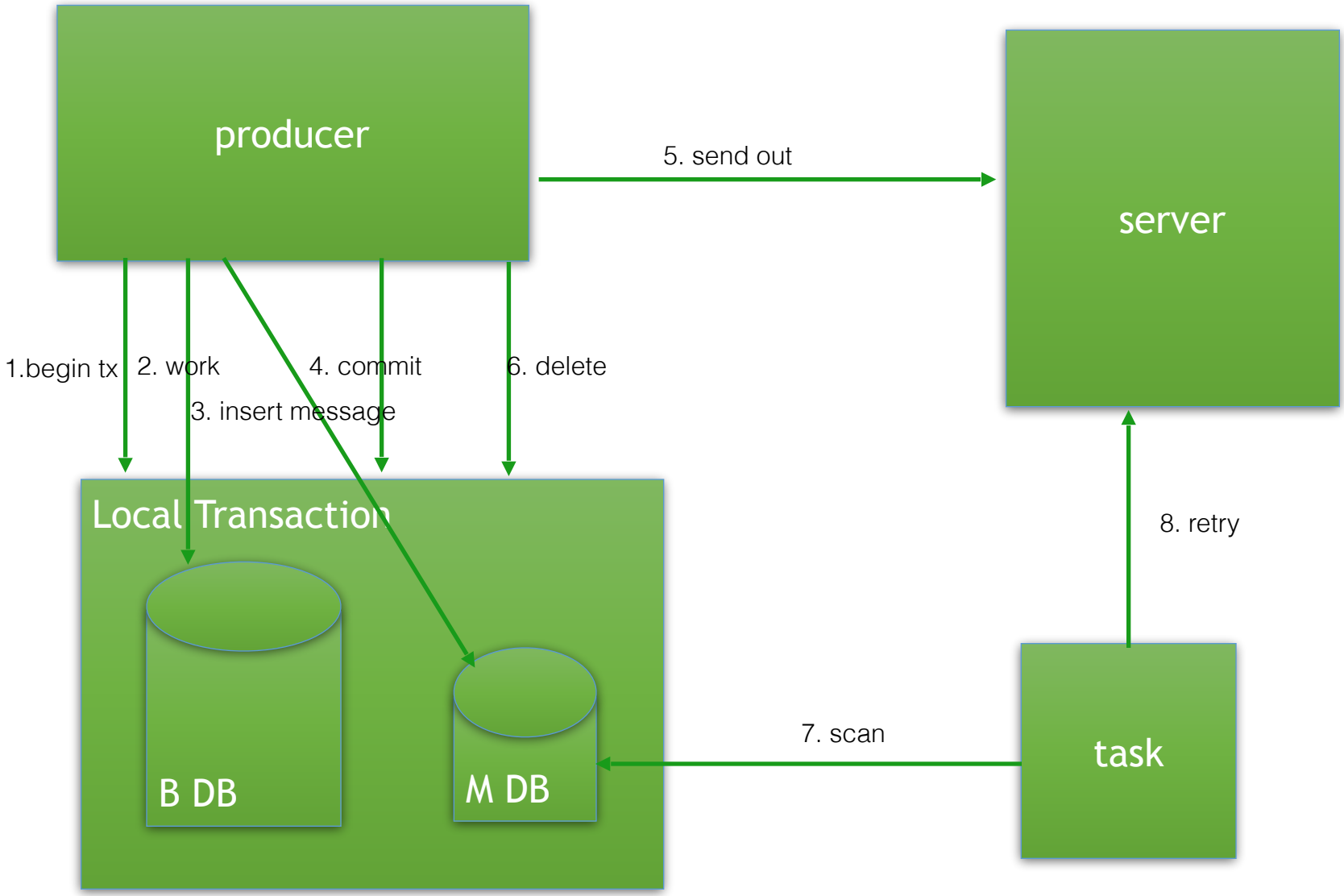
So many jobs……

Transactional Message Queue

```
@Transactional
public void pay(long orderId){
  payService.pay(orderId);
  producer.send(buildMessage(orderId));
}

public void onMessage(Message msg){
  orderServer.updateStatus(msg.getLong("orderId"));
}
```


DB Instance 3306
B DB1
B DB2
M DB
Local Transactional

```
@Transactional
public void pay(long orderId){
    payService.pay(orderId);

    //producer.send(buildMessage(orderId));

    //伪代码
    final Message payEvent = buildMessage(orderId);
    messageDAO.save(payEvent);
    postTransactionCommit(new Action(){
        @Override
        public void apply(){
            transport.send(payEvent);
        }
    });
}
```


producer
server
5. send out
1.begin tx
2. work
3. insert message
4. commit
6. delete
Local Transaction
B DB
M DB
8. retry
7. scan
task

服务端设计 — 存储模型

First Version

messages table

id	message id	content
1	19490488914	…
2	19477488902	…
3	19434488345	…

1 : N

send state table

id	message id	group	state
1	19490488914	oc	SENT
2	19490488914	flight	ACKED
3	19490488914	hotel	NACK

check send state job

Second Version

messages table

send state HMSET

id	message id	content
1	19490488914	...
2	19477488902	...
3	19434488345	...

	key	group	state
	19490488914	oc	SENT
		flight	ACKED
		hotel	NACK

scan messages

We want…

性能够好

成本够低

堆积能力

支持大量的topic

支持大量的consumer group

依赖少

他山之石

Kafka

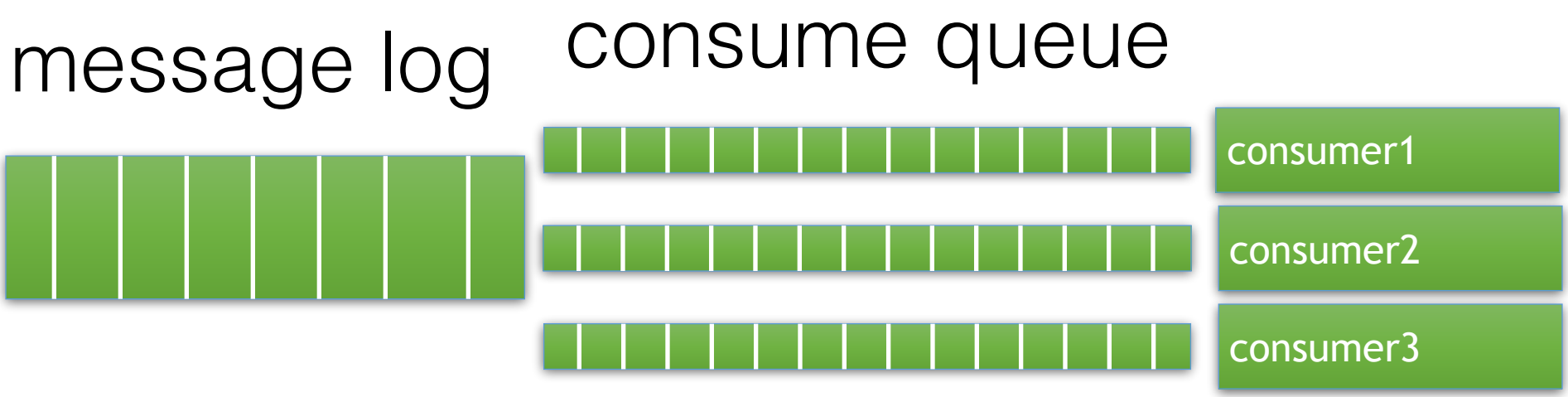
RocketMQ

topic

Log – 顺序文件

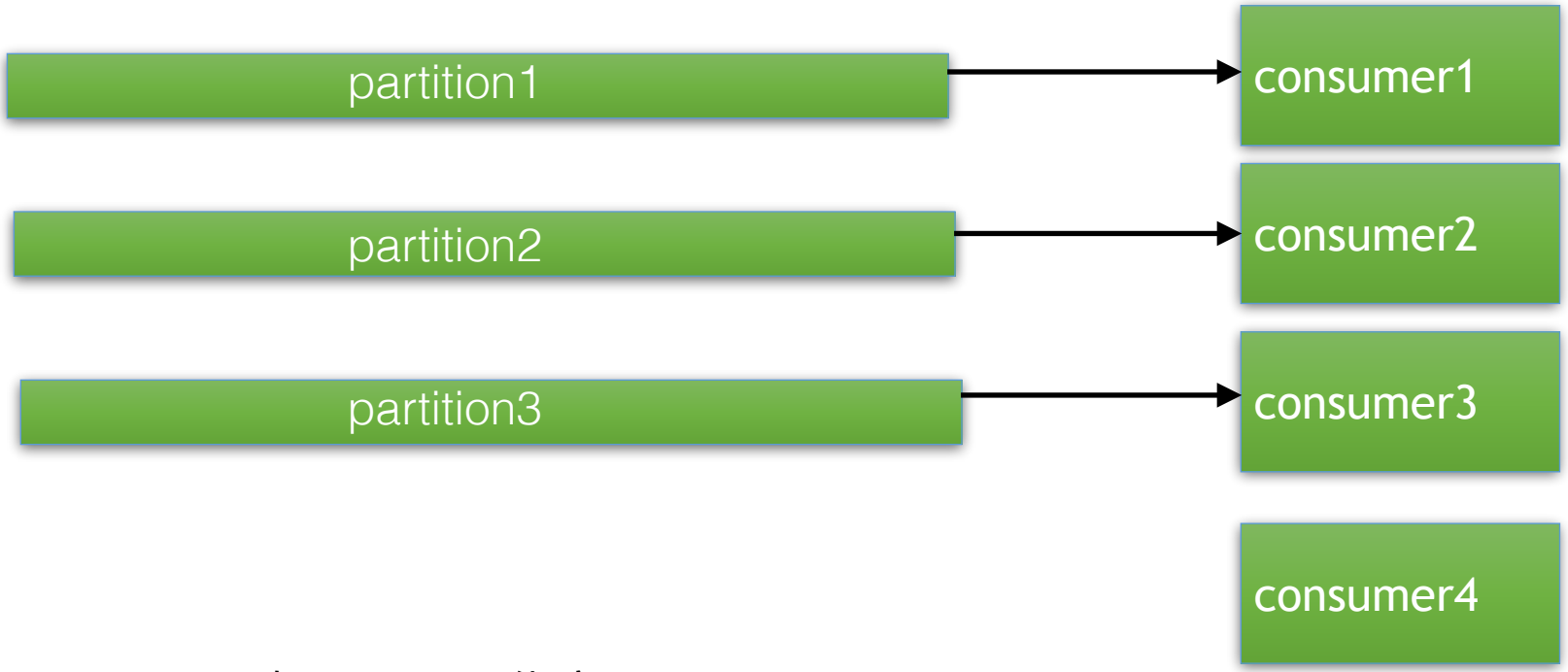
Offset – 消费进度

Kafka



RocketMQ

减少partition，降低随机写



partition与consumer绑定

busy

partition与consumer绑定

consumers

partition数与consumer个数成倍数关系

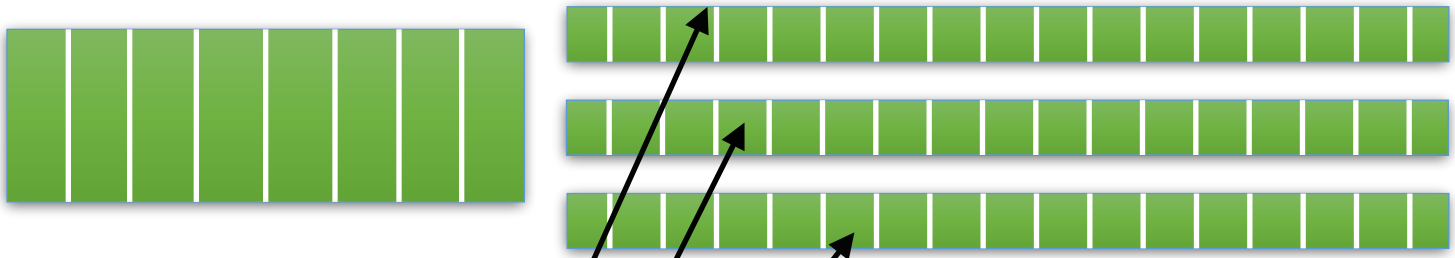

消息详情 .order.store.update / 28343.11
生产者
服务器
消费者 (54个)
消费记录
第1次
应用名:
[hotel_qta_order_store_provider]
发送端:
qunar.com
发送时间:
2017-12-17 16:12:40
状态：发送成功
ncn8 broker
应用名：[hotel_qta_order_store_provider]
订阅prefix：hotel.qta.order.store.update
订阅group：hotel.qta.store.oplog.send.hbase
消费成功：1次
重发
应用名：[des_internm]
订阅prefix：hotel.qta.order.store.update
订阅group：trade_group
消费成功：1次
重发
应用名：[sp_exchange_order]
订阅prefix：hotel.qta.order.store.update
订阅group：sp_exchange_order_group
消费成功：1次
重发
应用名：[m_ebooking_hos_server]
订阅prefix：hotel.qta.order.store.update
订阅group：HOS_INTEGRAL
消费成功：1次
重发
第1次
开始消费 qunar.com 2017-12-17 16:12:40
ping
消费成功 qunar.com 收到ack 2017-12-17 16:12:40

partition与consumer静态绑定

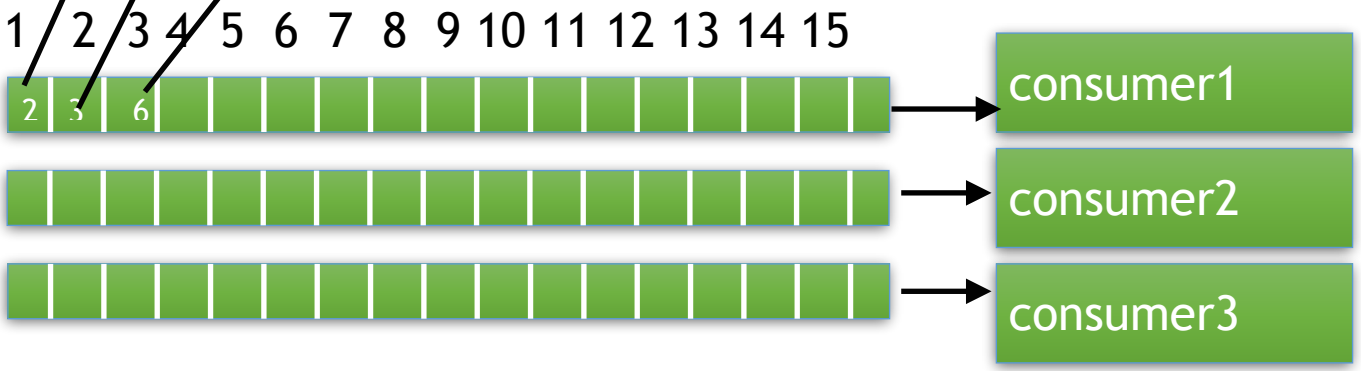
All problems in computer science can be solved by another level of indirection.

--David Wheeler

message log consume log



pull log



QMQ

QMQ

- consumer与”partition”动态绑定(pull log)
- 顺序文件
- 减少partition
- consumer个数任意增减
- 快速消化堆积的能力
- 堆积能力


消费积压---间隔2分钟 详情
1.6亿
30
11-22
20:45
2017-11-22
21:00
2017-11-22
21:15
2017-11-22
21:30
2017-11-22
21:45
2017-11-22
22:00
2017-11-22
22:15
2017-11-22
22:30
2017-11-22

QMQ

定时/延迟消息

消息轨迹

消息回溯

死信息队列

消费端集成幂等支持

...

Thank You

GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

扫码关注GIAC公众号

2017.thegiac.com