薄荷

薄荷Rails事务处理实践

张中南

2019/08/24

薄荷

About Me:

张中南 aka steamzhang

薄荷冰山团队技术负责人

RubyChina ID: sidekiq

薄荷

我们今天聊聊…

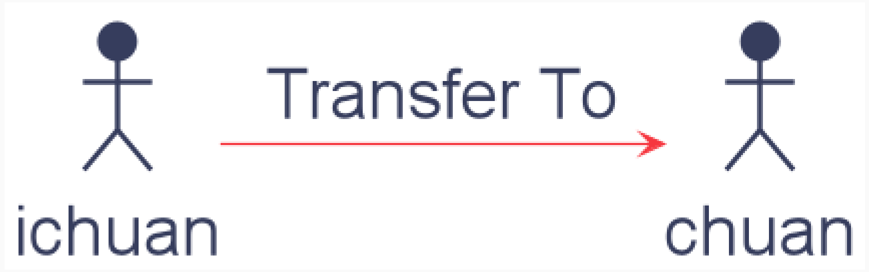
动账处理

秒杀活动

分布式事务

动账

Transfer Basic



```
def 恰烂钱
  ichuan = User.find_by_name 'ichuan'
  chuan = User.find_by_name 'chuan'

  ichuan.decrement money: 100
  chuan.increment money: 100
end
```

薄荷

Transfer Basic

kill -9 11092

```
def 恰烂钱
  ichuan = User.find_by_name 'ichuan'
  chuan = User.find_by_name 'chuan'

  ichuan.decrement money: 100
  chuan.increment money: 100
end
```

恰烂钱1.0 ...

```
def 恰烂钱v1_0
  ActiveRecord::Base.transaction do
    ichuan = User.find_by_name 'ichuan'
    chuan = User.find_by_name 'chuan'

    ichuan.decrement money: 100
    chuan.increment money: 100
  end
end
```

What if...

Transfer server instance

```
def 恰烂钱v1_0
  ActiveRecord::Base.transaction do
    ichuan = User.find_by_name 'ichuan'
    chuan = User.find_by_name 'chuan'

    ichuan.decrement money: 100
    chuan.increment money: 100
  end
end
```

In the meantime

```
chuan.increment money: 100
```

薄荷

What if...

Transfer server instance

```
def 恰烂钱v1_0
  ActiveRecord::Base.transaction do
    ichuan = User.find_by_name 'ichuan'
    chuan = User.find_by_name 'chuan'

    ichuan.decrement money: 100
    chuan.increment money: 100
  end
end
```

In the meantime

```
chuan.increment money: 100
```

Then...

```ruby
def 恰烂钱v2_0
  ActiveRecord::Base.transaction do
    chuan = User.find_by_name 'chuan'
    ichuan = User.find_by_name 'ichuan'

    chuan.with_lock do
      chuan.increment! money: 100
      ichuan.with_lock do
        ichuan.decrement! money: 100
      end
    end
  end
end
```

Then...

```ruby
def 恰烂钱v2_0
  ActiveRecord::Base.transaction do
    chuan = User.find_by_name 'chuan'
    ichuan = User.find_by_name 'ichuan'

    chuan.with_lock do
      chuan.increment! money: 100
      ichuan.with_lock do
        ichuan.decrement! money: 100
      end
    end
  end
end
```

Same as

```ruby
chuan.transaction do
  chuan.lock!
  # ...
end
```

Then...

```ruby
def 恰烂钱v2_0
  ActiveRecord::Base.transaction do
    chuan = User.find_by_name 'chuan'
    ichuan = User.find_by_name 'ichuan'

    chuan.with_lock do
      chuan.increment! money: 100
      ichuan.with_lock do
        ichuan.decrement! money: 100
      end
    end
  end
end
```

Same as

```ruby
chuan.transaction do
  chuan.lock!
  # ...
end
```

```sql
SELECT * FROM users WHERE id=1 FOR UPDATE
```

薄荷

当尝试解决问题时
通常会引入新的问题

Deadlock

```ruby
def 恰烂钱v2_0
  ActiveRecord::Base.transaction do
    chuan = User.find_by_name 'chuan'
    ichuan = User.find_by_name 'ichuan'

    chuan.with_lock do
      chuan.increment! money: 100
      ichuan.with_lock do
        ichuan.decrement! money: 100
      end
    end
  end
end
```

In the meantime

```ruby
ichuan.with_lock do
  ichuan.decrement! money: 100
  chuan.with_lock do
    chuan.increment! money: 100
  end
end
```

Solve deadlock using resource ordering

```
def transfer(from:, to:, amount:)
  pre, after, pre_method, after_method =
      from.id > to.id ?
          [from, to, :increment, :decrement] :
          [to, from, :decrement, :increment]

  pre.with_lock do
    pre.send pre_method, amount
    after.with_lock {after.send after_method, amount}
  end
end
```

→ Resource ordering

Conclusion

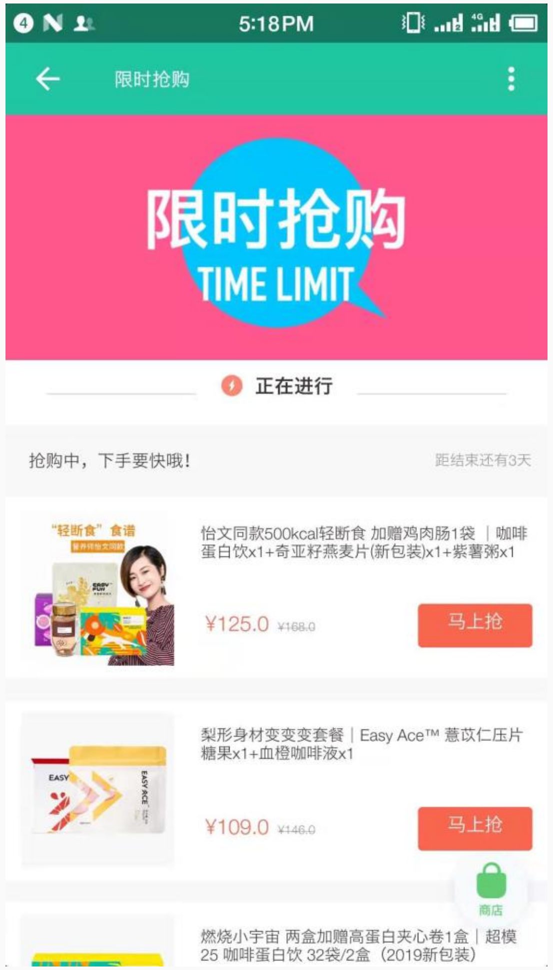
通过事务和锁可以有效防止状态不一致问题

事务嵌套很容易导致死锁

资源排序是一种解决死锁的思路

秒杀防超卖

秒杀防超卖


5:18PM

限时抢购

限时抢购
TIME LIMIT

正在进行

抢购中，下手要快哦!　　距结束还有3天

怡文同款500kcal轻断食 加赠鸡肉肠1袋 ｜咖啡蛋白饮x1+奇亚籽燕麦片(新包装)x1+紫薯粥x1

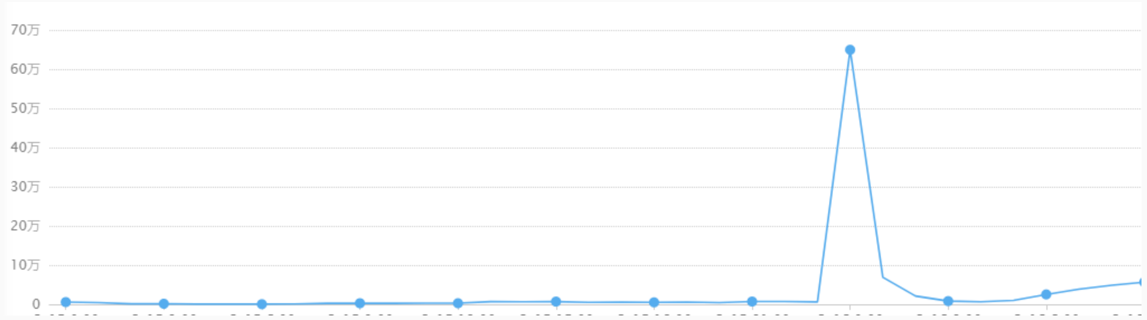
¥125.0 ¥168.0　　马上抢

梨形身材变变变套餐｜Easy Ace™ 薏苡仁压片糖果x1+血橙咖啡液x1

¥109.0 ¥146.0　　马上抢

商店

燃烧小宇宙 两盒加赠高蛋白夹心卷1盒｜超模25 咖啡蛋白饮 32袋/2盒（2019新包装）



方案对比

数据库锁-悲观锁

版本控制-乐观锁

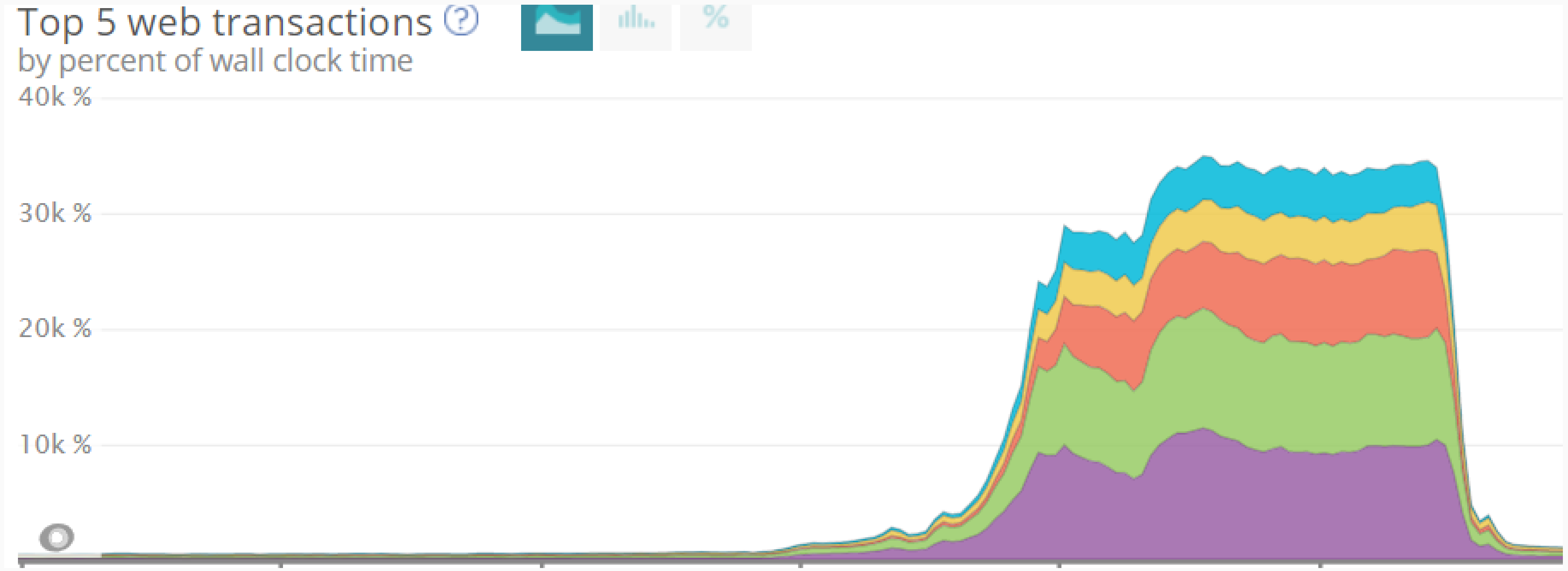
库存缓存

数据库锁

数据库锁

全服GIL

数据库锁

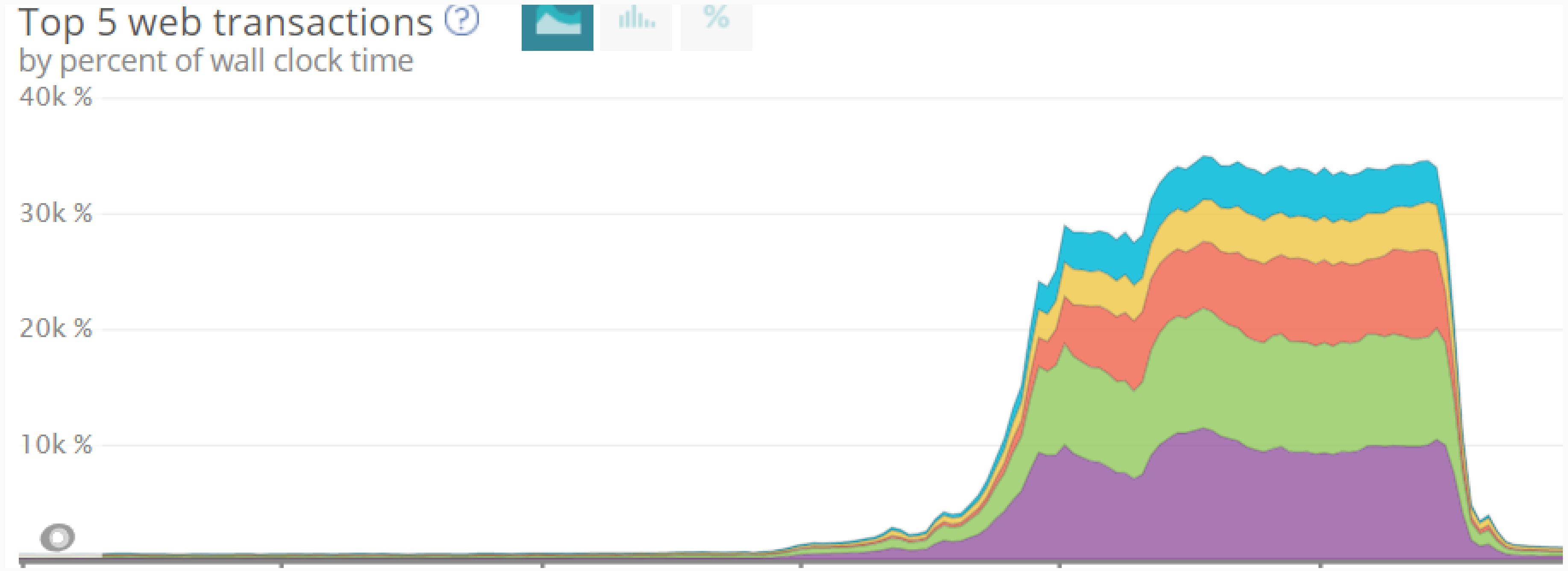

Top 5 web transactions
by percent of wall clock time
40k %
30k %
20k %
10k %

乐观锁

乐观锁

ABA问题

乐观锁



Top 5 web transactions
by percent of wall clock time
40k %
30k %
20k %
10k %

库存缓存

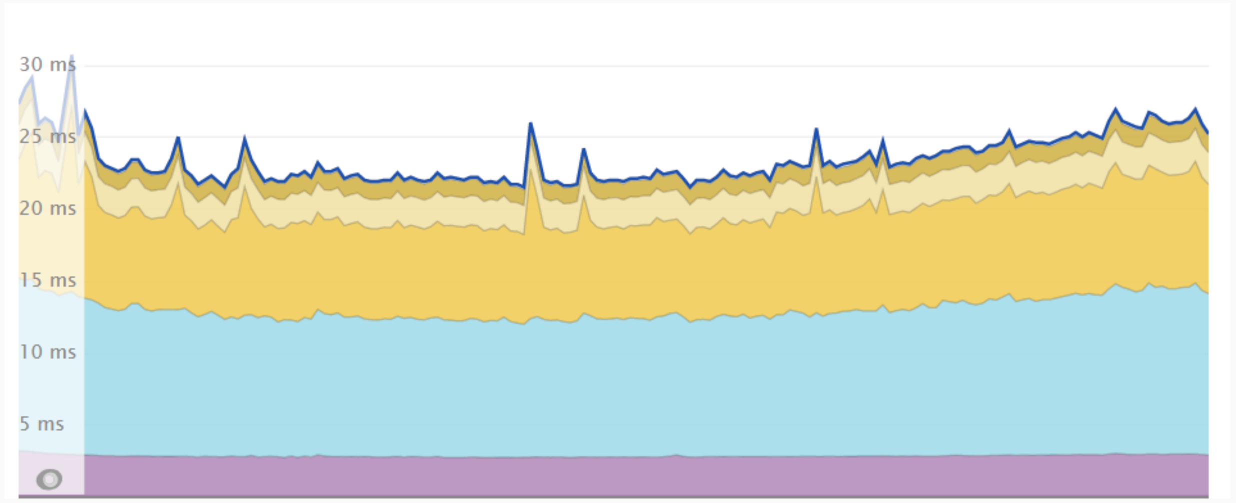
```ruby
class Goods < ApplicationRecord
  include Redis::Objects
  # ...
  counter name :stock
  # ...
end

def validate?(goods)
  value = goods.stock.decrement
  (value >= 0) ? true : false
end
```

库存缓存

```ruby
class Goods < ApplicationRecord
  include Redis::Objects
  # ...
  counter name :stock
  # ...
end

def validate?(goods)
  value = goods.stock.decrement
  (value >= 0) ? true : false
end
```



薄荷

Conclusion

悲观锁不适用于高并发情况

乐观锁不适用于高写入情况

缓存预减库存可以有效防止超卖

薄荷

Conclusion

悲观锁不适用于高并发情况

乐观锁不适用于高写入情况

缓存预减库存可以有效防止超卖

拓展思考😁

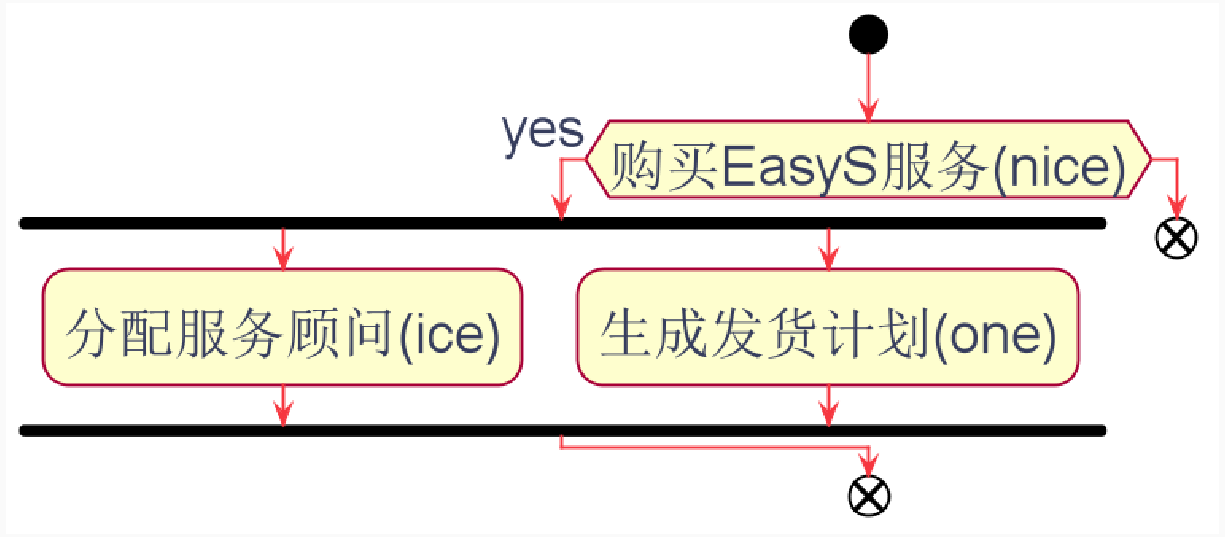
Redis挂了怎么办

Sentinel脑裂、多Master情况

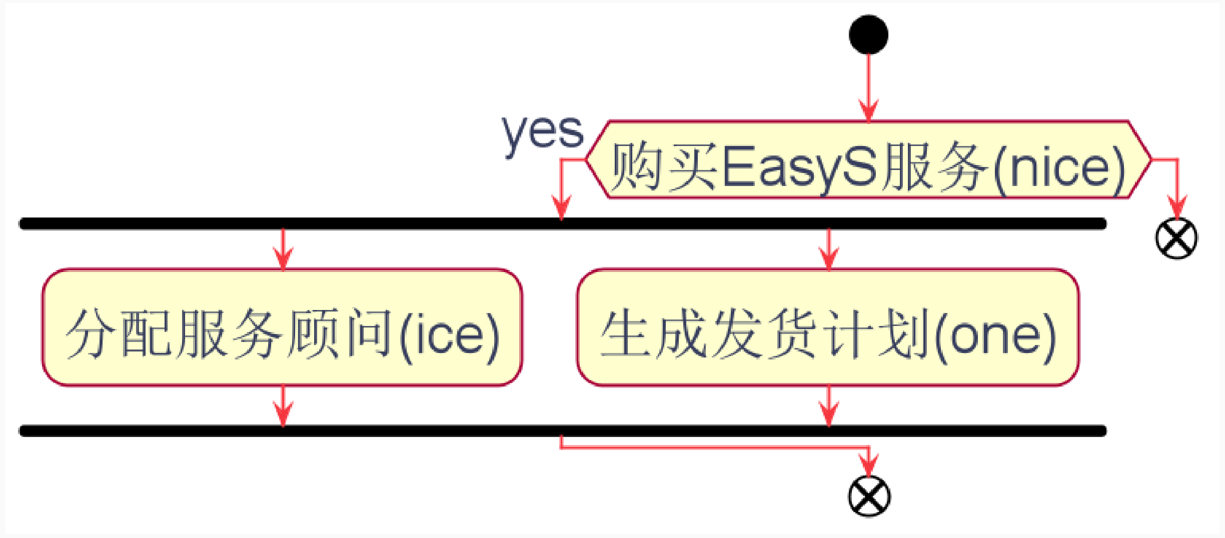
SSDB/PIKA可以做吗？

分布式事务

分布式事务



分布式事务

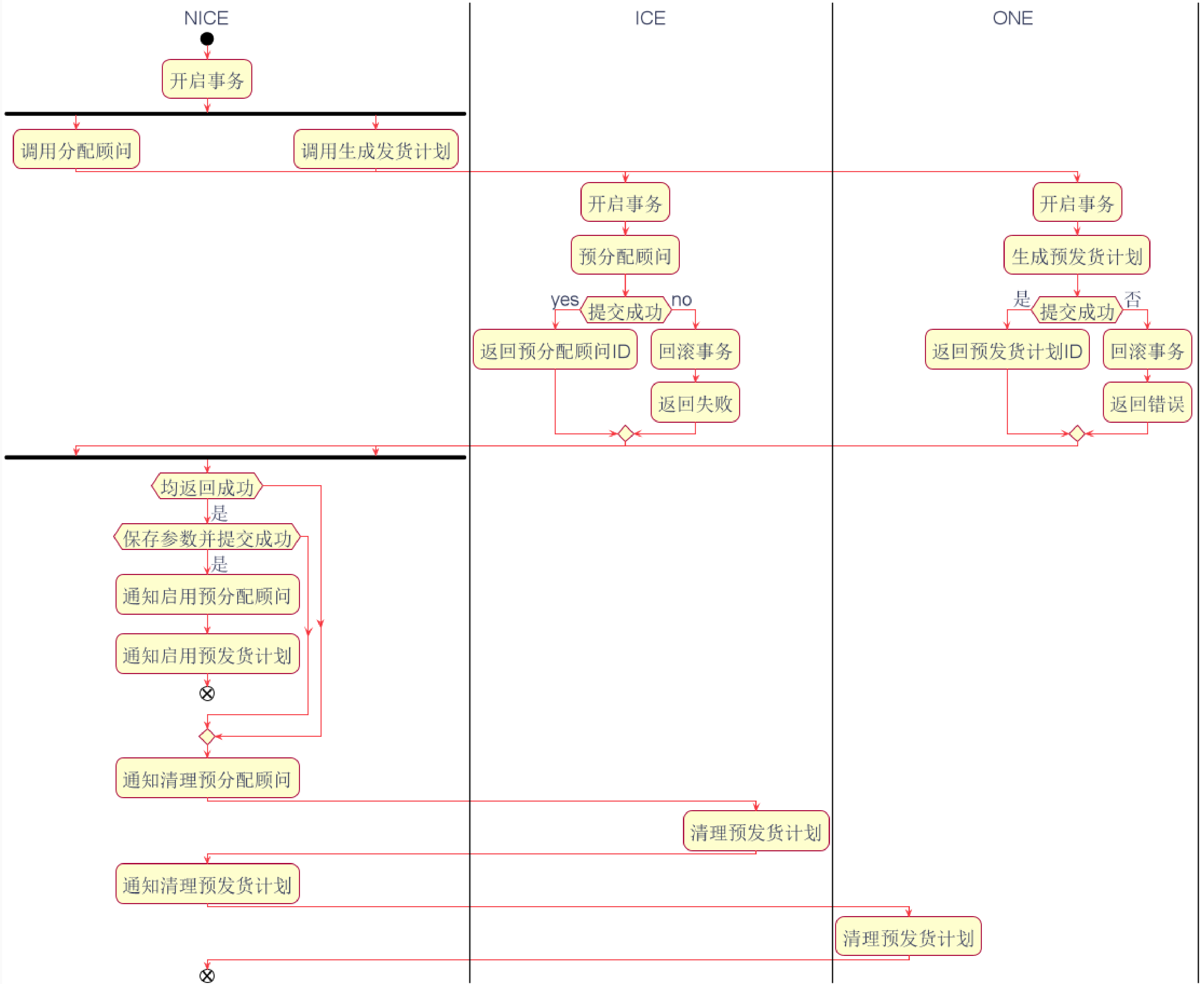


解决思路

串行执行

2PC

TCC


NICE
ICE
ONE
开启事务
调用分配顾问
调用生成发货计划
开启事务
预分配顾问
yes
提交成功
no
返回预分配顾问ID
回滚事务
返回失败
开启事务
生成预发货计划
是
提交成功
否
返回预发货计划ID
回滚事务
返回错误
均返回成功
是
保存参数并提交成功
是
通知启用预分配顾问
通知启用预发货计划
通知清理预分配顾问
清理预发货计划
通知清理预发货计划
清理预发货计划

Conclusion

先思考能否串行执行或通过业务改造进行

在TCC中，冻结-解冻模型是不错的实践

拥抱Java + RocketMQ生态😀

薄荷

万分感谢

薄荷

Q&A

谢谢大家