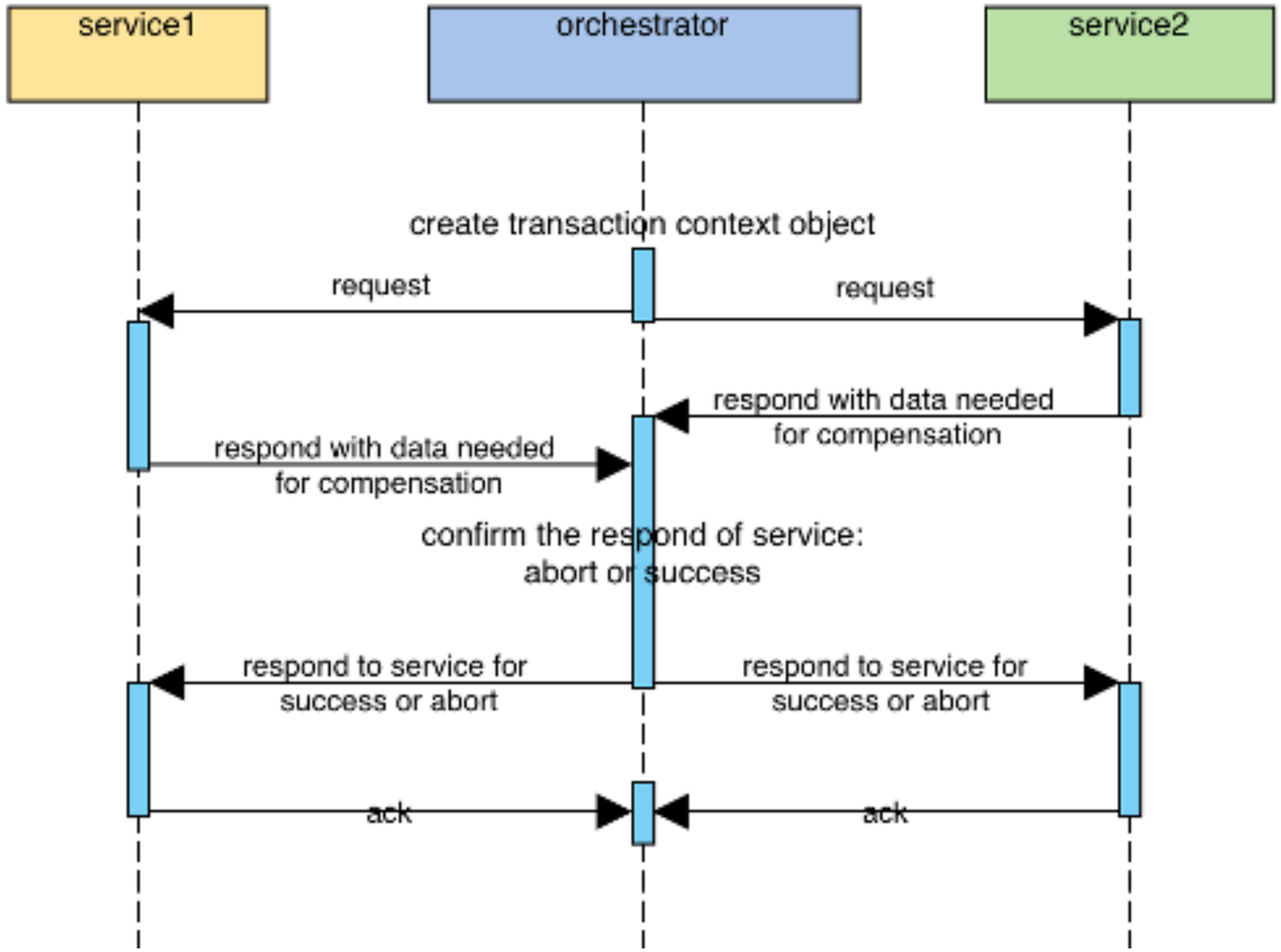
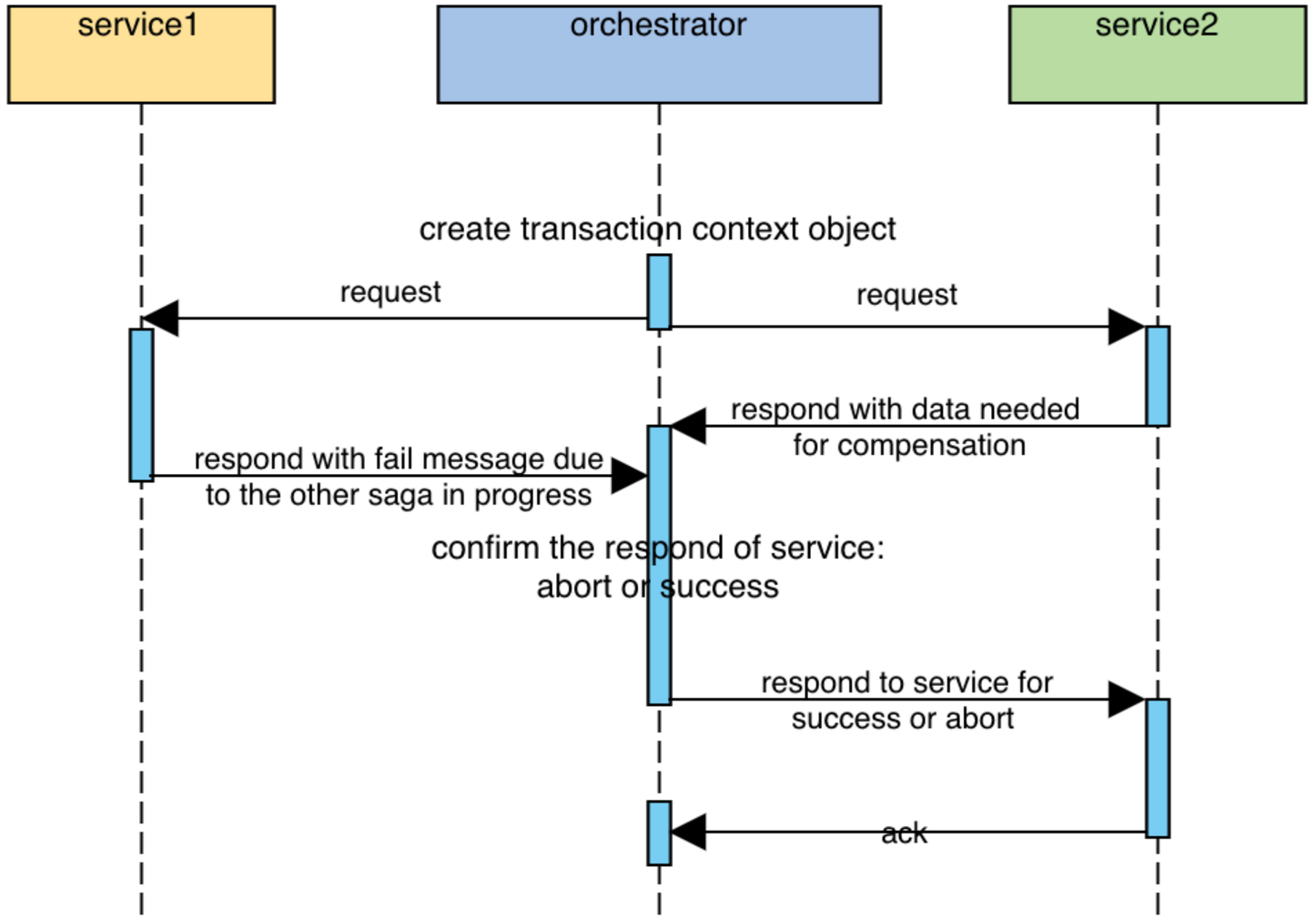
mqtt sagas

lock and no lock



lock and no lock



lock and no lock

```
constructor(transactionId, transientIdList) {
    this.transactionId = transactionId;

    this.services = [
      {
        serviceId: 'service1',
        state: 0,  // -1 = fail,  1 = success
        ack: false,
        reject: false,
        transientId: '',
        compensate:[] // 補償操作
      },
      {
        serviceId: 'service2',
        state: 0,
        ack: false, // -1 = fail,  1 = success
        reject: false,
        transientId: '',
        compensate:[] // 補償操作
      }
    ]
•
```

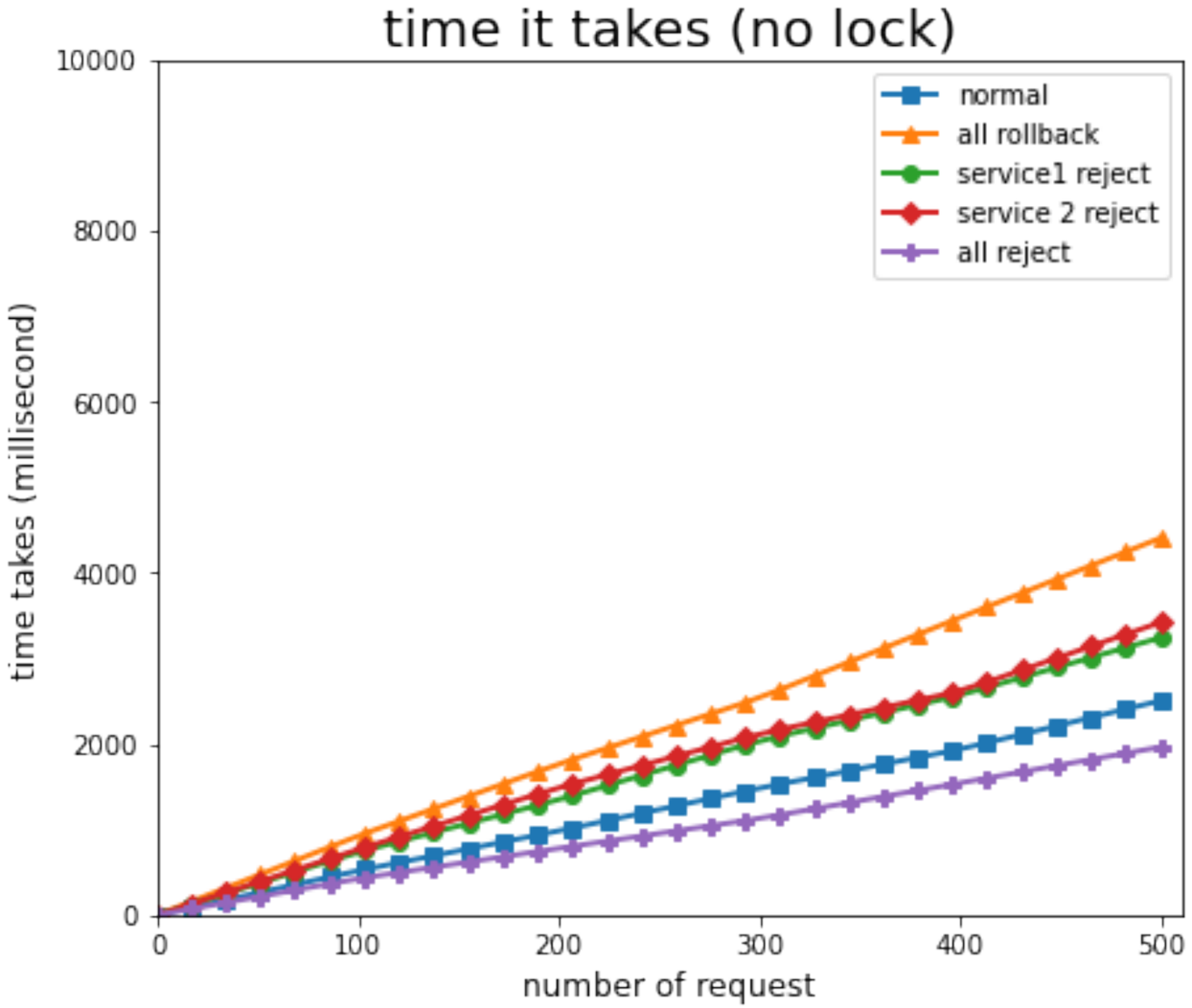
short circuit

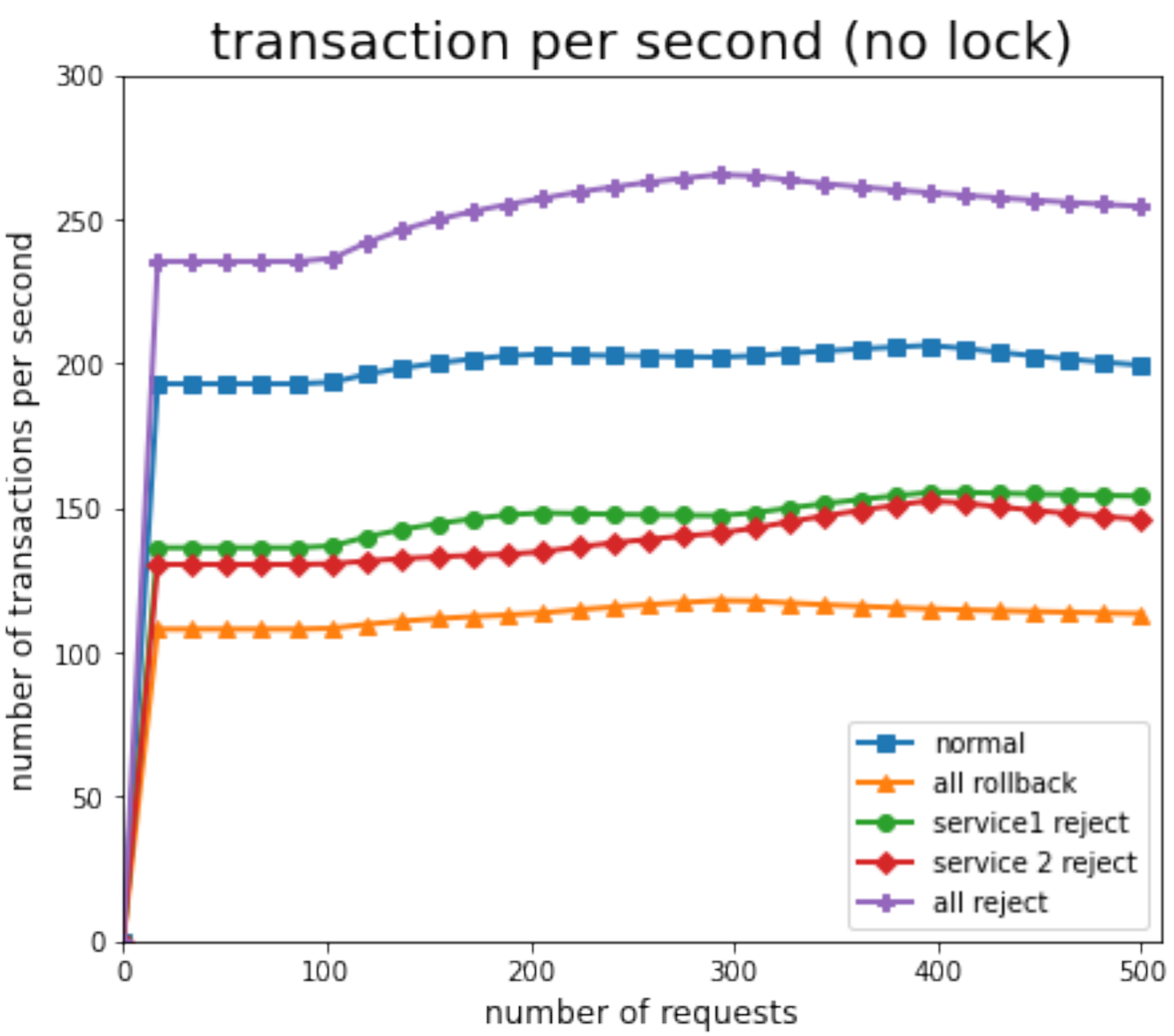
```
constructor(transactionId, transientIdList) {
    this.transactionId = transactionId;

    this.services = [
      {
        serviceId: 'service1',
        state: 0,  // -1 = fail,  1 = success
        ack: false,
        fail: false,
        reject: false,
        transientId: '',
        compensate:[] // 補償操作
      },
      {
        serviceId: 'service2',
        state: 0,
        ack: false, // -1 = fail,  1 = success
        fail: false,
        reject: false,
        transientId: '',
        compensate:[] // 補償操作
      }
```

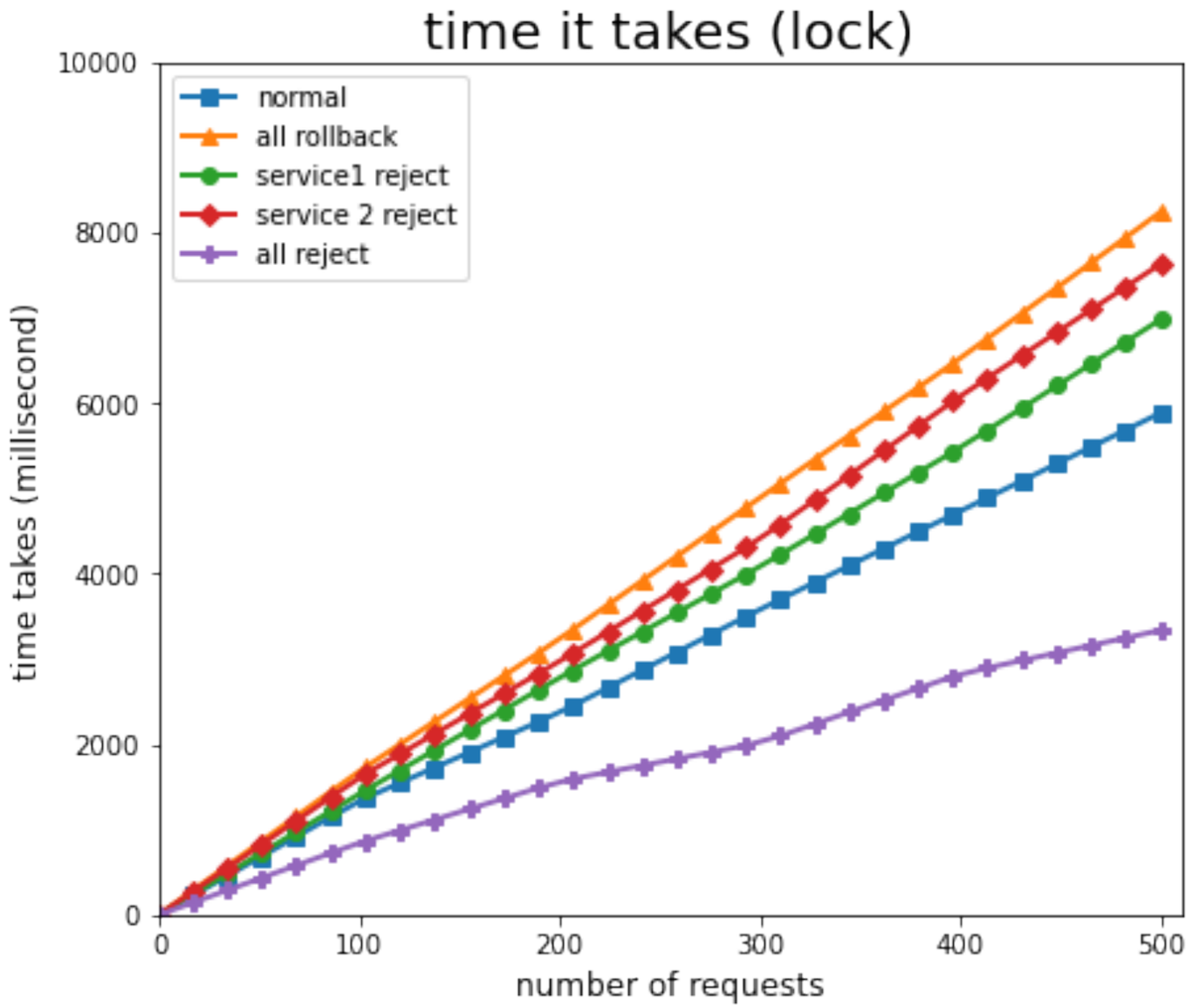
•

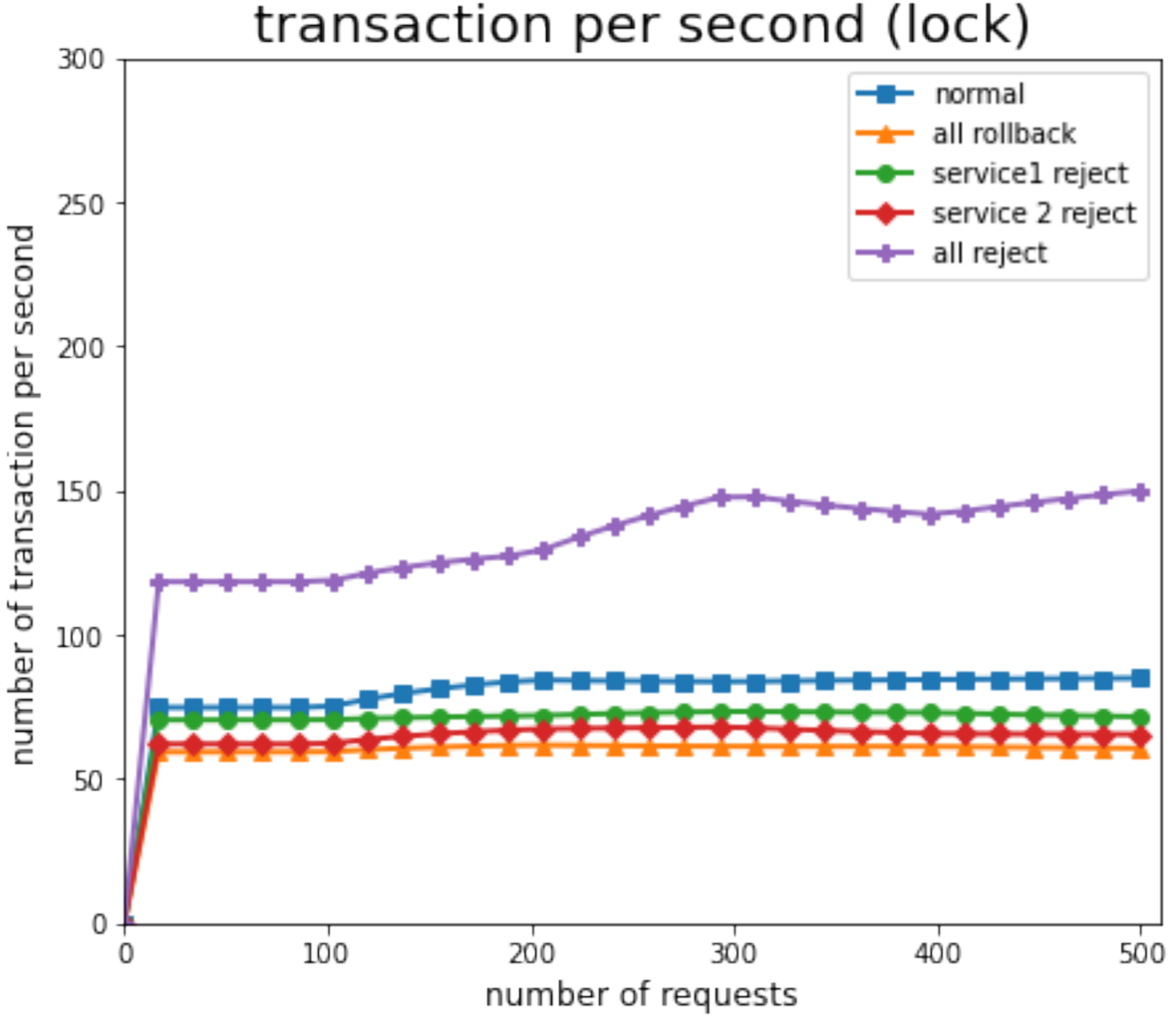
no lock experiment



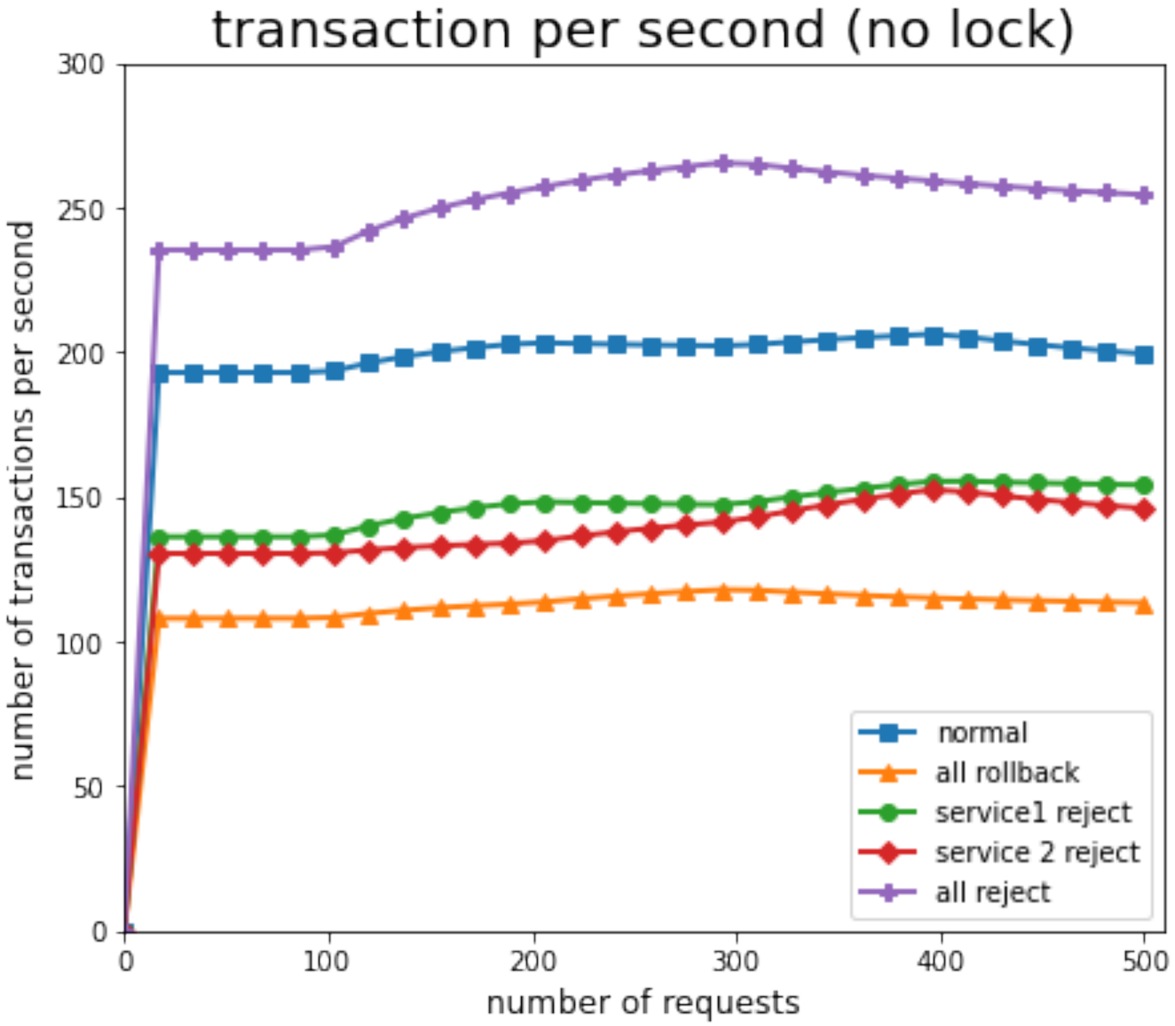


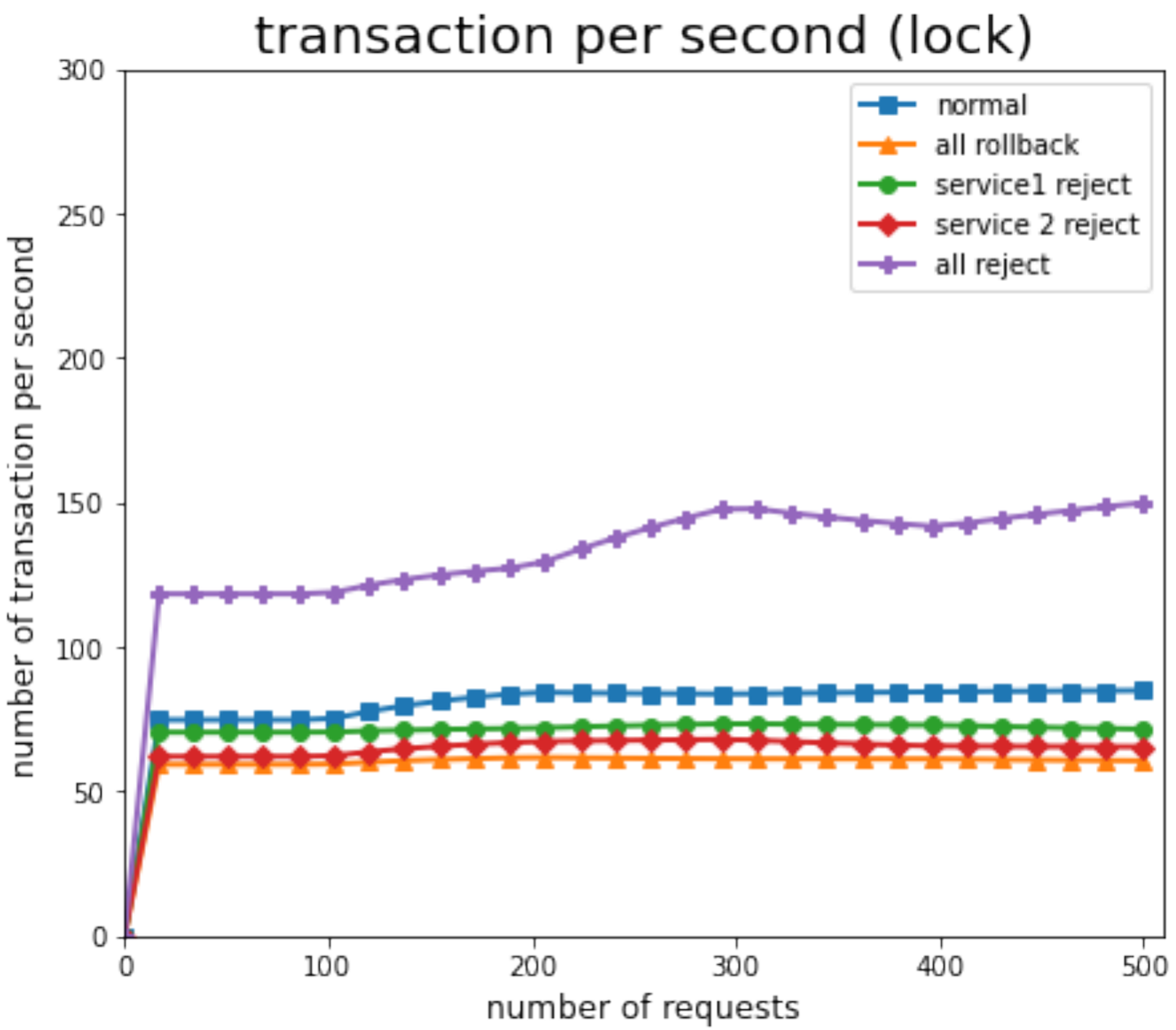
lock experiment



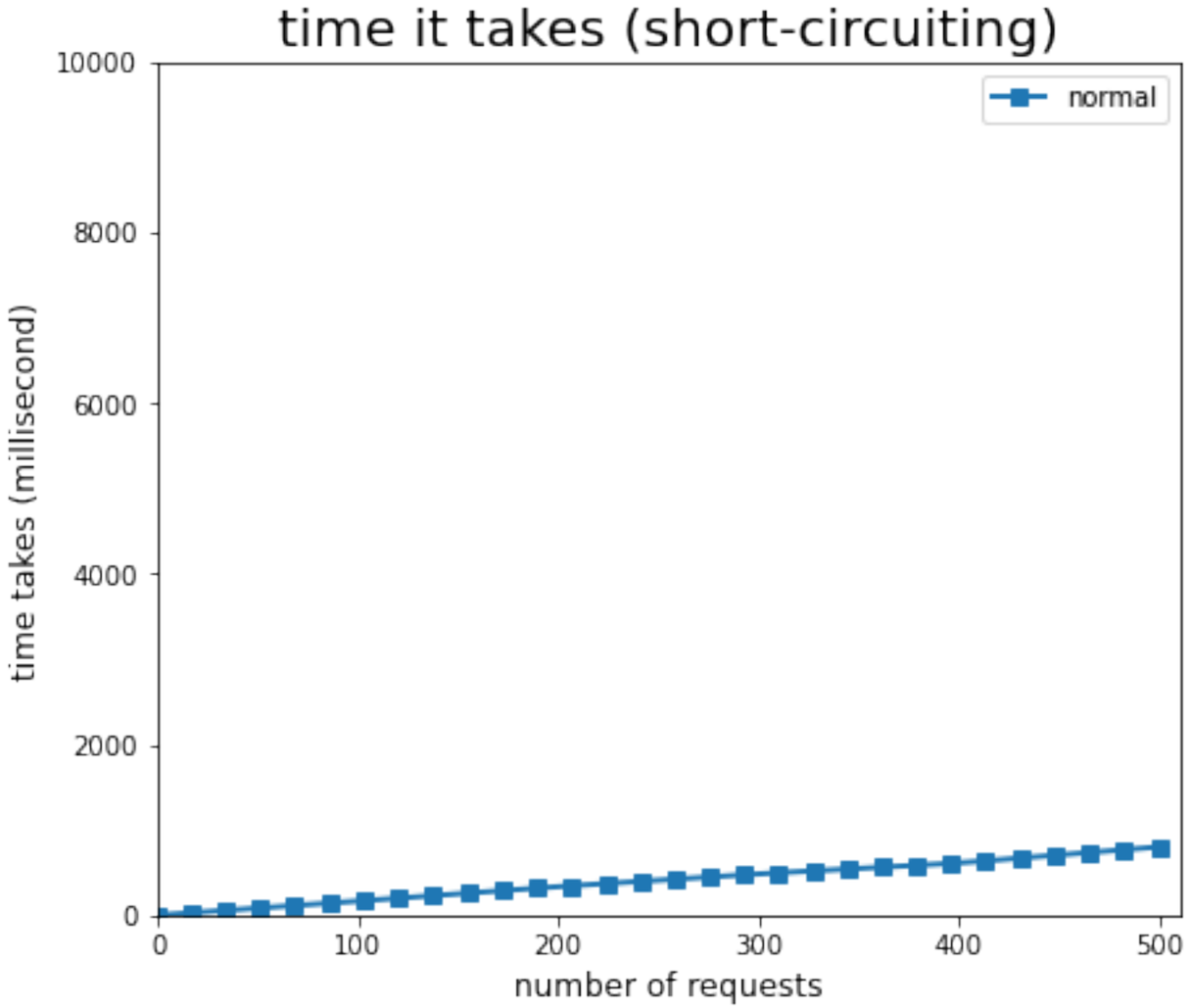


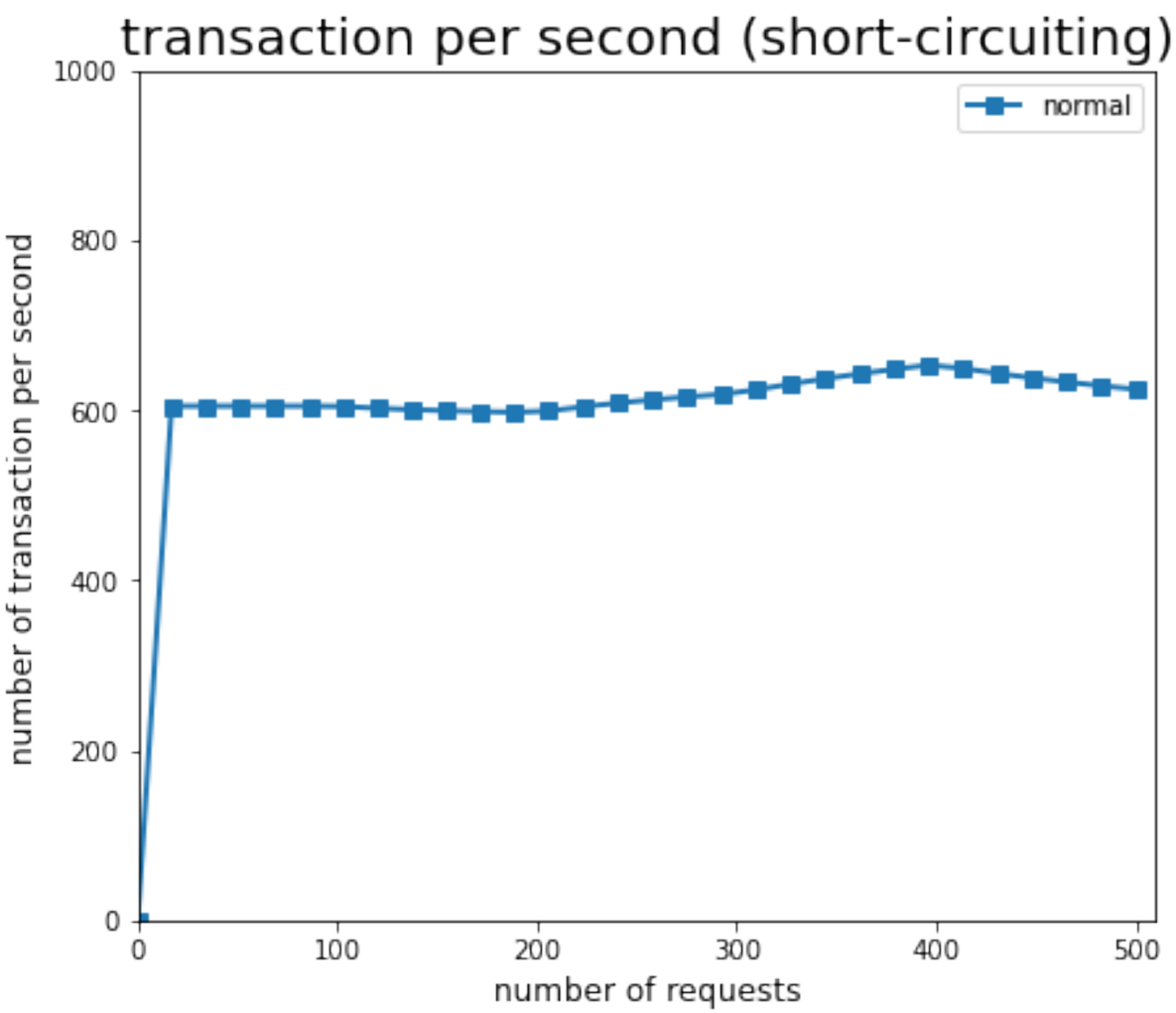
comparison


transaction per second (no lock)
number of transactions per second
number of requests
0
50
100
150
200
250
300
0
100
200
300
400
500
normal
all rollback
service1 reject
service 2 reject
all reject


transaction per second (lock)
number of transaction per second
number of requests
0
50
100
150
200
250
300
0
100
200
300
400
500
normal
all rollback
service1 reject
service 2 reject
all reject

short circuit





paper

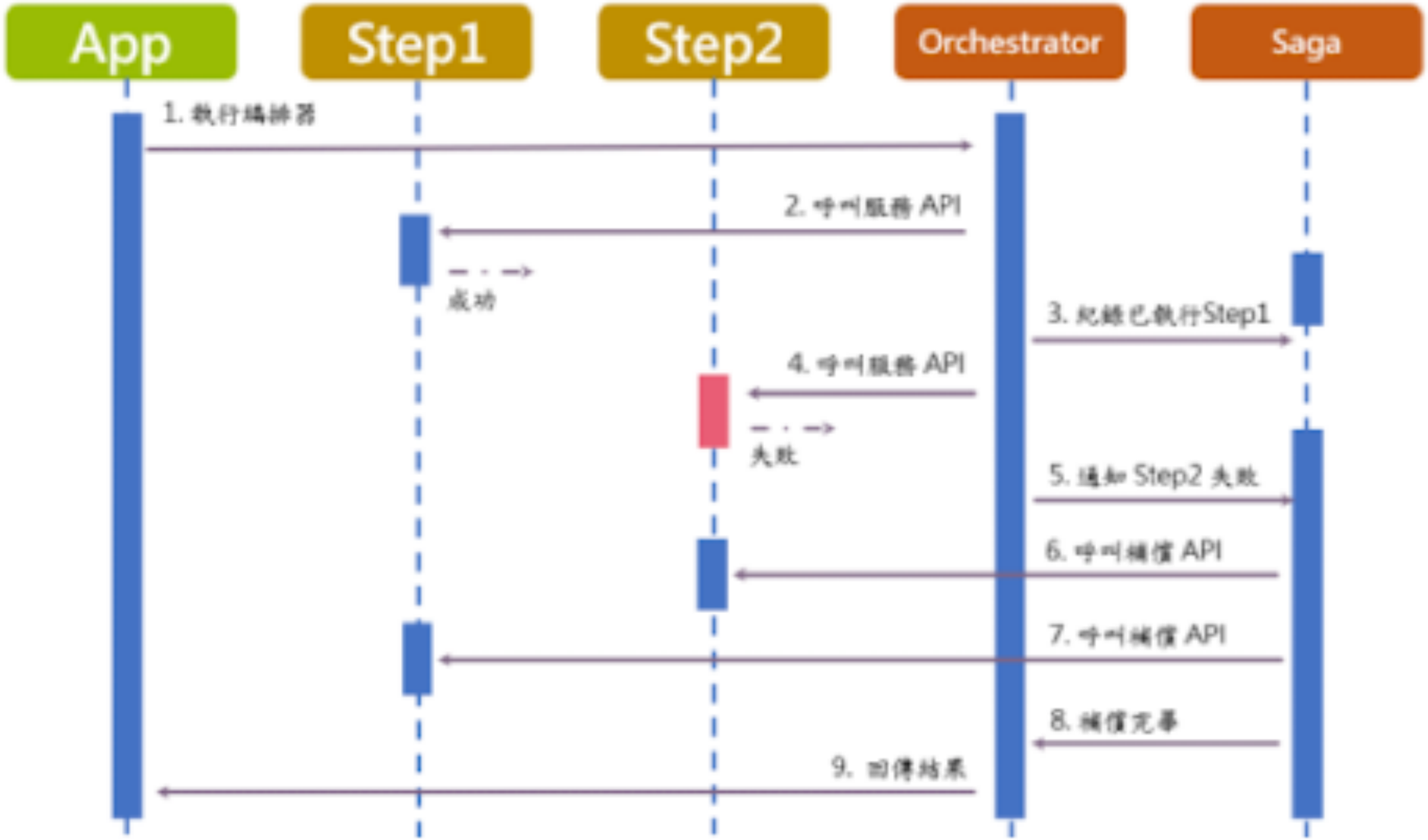


圖 7. Saga 補償機制

confirm ?

案例	執行細節
新建訂單成功