交易前台ACE---OMS（v0.0.1）

# 目录

[目录 1](#_Toc39962372)

[修订（开发）记录 3](#_Toc39962373)

[一．OMS总体设计 4](#_Toc39962374)

[二．OMS周边服务交互图 5](#_Toc39962375)

[三．OMS内部组件设计 6](#_Toc39962376)

[3.1 状态机 6](#_Toc39962377)

[3.2 异步持久化 6](#_Toc39962378)

[3.3 缓存队列 7](#_Toc39962379)

[3.4 线程池 7](#_Toc39962380)

[四．OMS各功能设计 8](#_Toc39962381)

[4.1 服务初始化（日切、服务重启） 8](#_Toc39962382)

[4.2 委托/撤单 8](#_Toc39962383)

[4.3 异步持久化 10](#_Toc39962384)

[4.4 异常恢复 11](#_Toc39962385)

[4.5 订单查询 11](#_Toc39962386)

[4.6 数据迁移备份 11](#_Toc39962387)

[五．对外服务接口 12](#_Toc39962388)

[六．数据库设计 13](#_Toc39962389)

[6.1 订单表（ace\_order） 13](#_Toc39962390)

[6.2 订单历史表（ace\_order\_his） 14](#_Toc39962391)

[6.3 成交回报表（ace\_execution） 15](#_Toc39962392)

[6.4 成交回报历史表（ace\_execution\_his） 15](#_Toc39962393)

[6.5 异常任务表（ace\_exception\_task） 15](#_Toc39962394)

# 修订（开发）记录

| 日期 | 版本 | 修订说明 | 修订者 |
| --- | --- | --- | --- |
| 2020.05.09 | v0.0.1 | 1. ACE-OMS详细设计起草 | 姜瀚 |
|  |  |  |  |

# 一．OMS总体设计

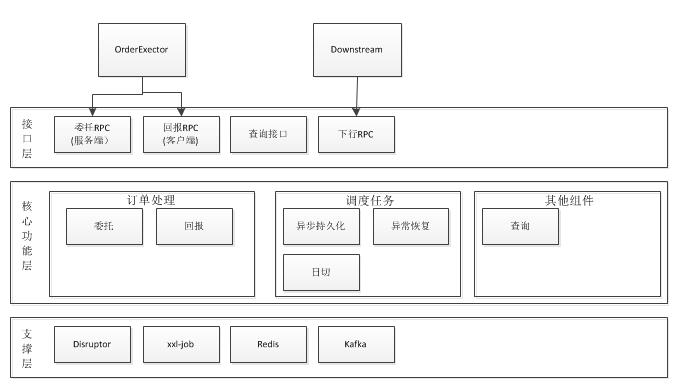


图1.1 ACE-OMS总体设计

Order Manager Service（OMS）订单管理服务是ACE中的一大服务，担负以下职责：

1. 统一管理委托：交易中台的OrderExector与StrategyContainer是一对一关系的，所以OrderExector没有全局的视角；而OMS与OrderExector是多对一的，所有的订单都要经过OMS集群，从而有全局的视角。
2. 统一管理回报：同上。
3. 管理持仓和现金

# 二．OMS周边服务交互图

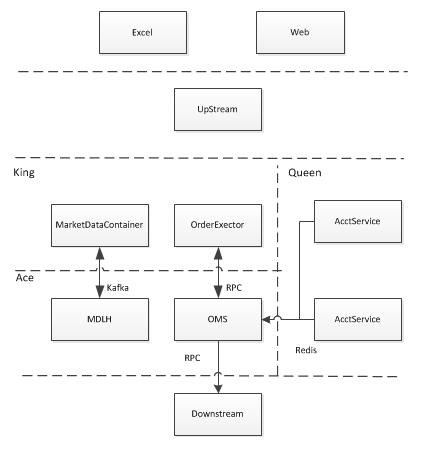


图2.1 OMS与周边模块的交互

# 三．OMS内部组件设计

## 3.1 状态机

订单状态机，包括Init(初始化订单)、New（新订单未报）、Pending(新订单已报)、Part-Exec（部分成交）、Full-Exec（完全成交）、Rejected（订单被拒绝）、Withdrawing-trade（已报交易所的订单撤单）、Withdraw（订单未成交，全部撤回）。

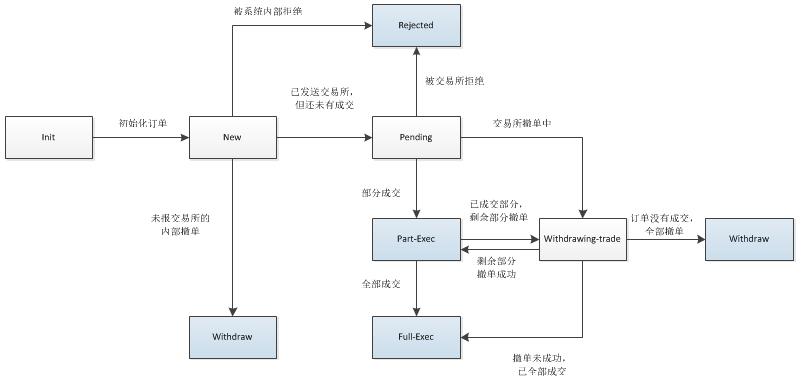


图3.1 订单状态机

深色为终态，除此以外都是中间态。

## 3.2 异步持久化

OMS采用异步持久化，即收到订单、成交回报，只在内存中处理并写入redis，另起线程做持久化mysql的操作。

用定时任务来执行mysql的持久化

## 3.3 缓存队列

两个disruptor队列缓存：order队列和executor队列

## 3.4 线程池

线程1：持久化数据库数据

线程2：市场开放时，读取order队列，RPC downstream进行委托下单

线程3：读取executor队列，并RPC交易中台发送交易回报

线程4：异常恢复

# 四．OMS各功能设计

## 4.1 服务初始化（日切、服务重启）

Monitor每日日终会强制所有服务下线重启（或脚本定时任务），OMS重启后，做以下事情：

1. 重新从全局配置表获取BussinessDate
2. 将Redis中的order和execution表的数据非当日的交易全部删除
3. 将redis中的当天的order和execution表导入内存（即重启恢复）

## 4.2 委托/撤单

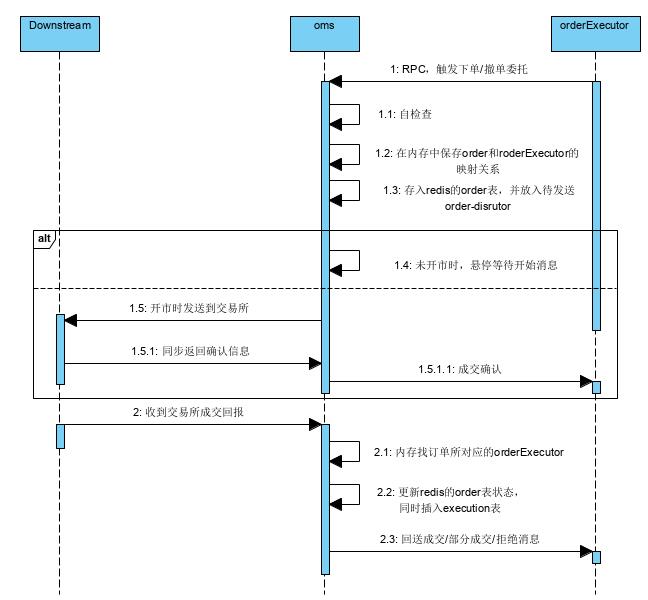


图4.1 委托/撤单时序图

收到交易中台的订单后，做如下处理：

1. 写入order缓存队列
2. OMS判断当前是否是开市时间，如果已开市，轮询order线程持续等待；如果已开市，order线程调起缓存中的order。
3. 检查order是否已在缓存中存在，如果已经存在，即判断为重复单（可能是上游KING的bug），报警并丢弃；如果不存在加入到内存的order-OrderExecutor的映射，以便回报和交易确认的回调。
4. 调downstream，更改状态机为New，并写入redis；如果发生异常，插入异常任务表，（对应order的状态变为exeception？，待讨论）
5. 写入持久化的order队列

收到交易中台的撤单请求：

1. 写入order缓存队列
2. 找到内存中的订单（如果没有，要么是丢单，要么是服务重启，之前的数据持久化失败了，待讨论如何处理）
3. 如果是未报状态，直接更新状态为Withdraw
4. 如果是已报状态，向downstream发送撤单申请，如果发生异常，插入异常任务表（对应order的状态变为exeception？，待讨论）
5. 如果是完全成交，无需处理
6. 调交易中台的rpc通知撤单结果
7. 写入持久化的order队列

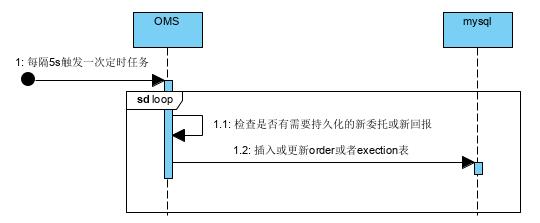
收到downstream的订单确认（已发送交易所，或者交易所拒绝，或部分成交）：

1. 写入execution缓存队列

2. execution线程调起execution缓存的结果

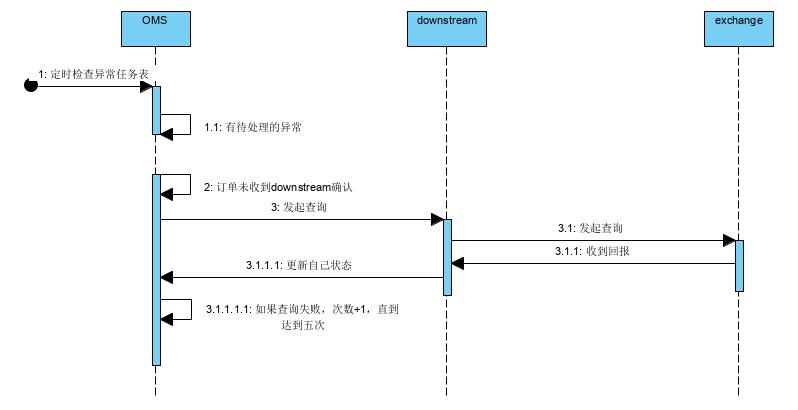
1. 如果是订单确认：更新状态为Pending，发送交易中台（也可不发送，待商量），更新redis
2. 如果是订单拒绝：更新状态为Rejected，发送交易中台，更新redis
3. 如果是成交报文：更新状态为full或part，发送交易中台，更新redis
4. 写入持久化的execution队列

## 4.3 异步持久化



1. 配置xxl-job任务，也可以用jdk的定时线程
2. 每5秒（配在全局参数表）吊起，查看待持久化的order和execution队列是否为空。
3. 如果不为空，则进行插入数据库（新委托、新成交、撤单），或者更新数据库（部分成交）操作。

## 4.4 异常恢复



只处理长时间New的委托，即OMS已经调过downstream，但一直没有回报的委托

1. 异常任务线程检查内存订单表或者redis，找到超过5s仍未变为pending的订单，插入异常任务表。
2. 扫描异常任务表，向downstream发起查询，如果依然失败，对应异常任务次数加1；如果downstream告知未找到该笔订单，不做重新委托的操作，通知交易中台；如果downstream返回该笔委托信息，更新oms的状态
3. 如果异常任务恢复超过3次依然失败，通知交易中台

## 4.5 订单查询

## 4.6 数据迁移备份

使用FAMS框架的迁移插件（待开发）

# 五．对外服务接口

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 描述 | 调用方 | 参数描述 |
| NewOrder | 提供给上游的普通委托回调接口 | KING | 参考order表 |
| ExecutionOrder | 提供给下游的成交回报回调接口 | Downstream | 参考execution表 |
| QueryOrder | 查询当日委托、历史委托 | Upstream、KING | 自行设计 |
| QueryExecution | 查询成交、历史成交 | Upstream、KING | 自行设计 |
|  |  |  |  |

Ps：未来还需要支持组合单

# 六．数据库设计

## 6.1 订单表（ace\_order）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必填 | 描述 |
| id | bigint | 是 | 主键 |
| order\_id | varchar(64) | 是 | 订单编号 |
| account | varchar(32) | 是 | 交易账户 |
| security\_code | varchar(16) | 是 | 证券标的 |
| exch | varchar(8) | 是 | 交易市场 |
| order\_status | varchar(16) | 是 | 订单状态机中的状态：  Init(初始化订单)、New（新订单未报）、Pending(新订单已报)、Part-Exec（部分成交）、Full-Exec（完全成交）、Rejected（订单被拒绝）、Withdrawing-trade（已报交易所的订单撤单）、Withdraw（订单未成交，全部撤回） |
| order\_reject\_code | varchar(16) | 否 | 填写拒单代码：  交易所拒单：EXCH\_REJECT  OMS拒单：OMS\_REJECT |
| order\_reject\_reason | varchar(64) | 否 | 拒单具体原因 |
| trade\_time | datetime | 是 | 交易时间 |
| order\_executor | varchar(32) | 是 | 订单来源名称，对应king的orderexecutor组件。用于回报等信息 |
| strategy\_container | varchar(32) | 否 | 如果是策略单，需要 |
| order\_price | bigdecimal(7,2) | 是 | 委托价格 |
| order\_volume | long | 是 | 委托量，以股为单位 |
| order\_type | varchar(16) | 是 | 订单类型：limit、fak、fok等 |
| order\_multiply | long | 是 | 证券乘数，例如1手=100股，此处为100 |
| order\_direction | varchar(8) | 是 | 委托方向：BUY/SELL/LONG/SHORT |
| exec\_price | bigdecimal(7,2) | 否 | 成交均价 |
| exec\_volume | long | 否 | 成交数量 |
| exec\_status | varchar(16) | 是 | 订单执行状态：  OPEN：等待执行，处于new、pengding、part-exec、withdraw-trade的订单，即还未完全执行结束的委托  CLOSE：执行完成，处于full-exec、Rejected、withdraw的订单，即已经结束的委托 |
| withdraw\_volume | long | 否 | 撤单数量 |
| modify\_time | datetime | 是 | 修改时间 |
| create\_time | datetime | 是 | 创建时间 |

## 6.2 订单历史表（ace\_order\_his）

同订单表（ace\_order）

## 6.3 成交回报表（ace\_execution）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必填 | 描述 |
| id | bigint | 是 | 主键 |
| order\_id | varchar(64) | 是 | 订单编号 |
| account | varchar(32) | 是 | 交易账户 |
| security\_code | varchar(16) | 是 | 证券标的 |
| exch | varchar(8) | 是 | 交易市场 |
| exec\_time | datetime | 是 | 成交回报时间 |
| exec\_price | bigdecimal(7,2) | 否 | 执行价格，如果是撤单的成交，可能没有价格 |
| exec\_volume | long | 是 | 成交数量 |
| withdraw\_flag | varchar(2) | 是 | 撤单标志。默认为N |
| modify\_time | datetime | 是 | 修改时间 |
| create\_time | datetime | 是 | 创建时间 |

## 6.4 成交回报历史表（ace\_execution\_his）

同成交回报（ace\_execution）

## 6.5 异常任务表（ace\_exception\_task）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必填 | 描述 |
| id | bigint | 是 | 主键 |
| order\_id | varchar(64) | 是 | 订单编号 |
| account | varchar(32) | 是 | 交易账户 |
| security\_code | varchar(16) | 是 | 证券标的 |
| exch | varchar(8) | 是 | 交易市场 |
| order\_status | varchar(16) | 是 | 订单状态机中的状态：  Init(初始化订单)、New（新订单未报）、Pending(新订单已报)、Part-Exec（部分成交）、Full-Exec（完全成交）、Rejected（订单被拒绝）、Withdrawing-trade（已报交易所的订单撤单）、Withdraw（订单未成交，全部撤回） |
| count | int | 是 | 重试次数，初始为0 |
| modify\_time | datetime | 是 | 修改时间 |
| create\_time | datetime | 是 | 创建时间 |