测试报告

成员：秦嘉余 赖烨文 刘永鹏 孙文戈 张城铨 蒋梓栩

日期： 2022年6月4日

引言

编写目的

为了发现和报告南大测试中心后端的错误和缺陷。通过测试，确保本系统的功能、互操作性等符合软件的设计要求，满足用户的使用要求。通过分析错误产生的原因和错误的分布特征，可以帮助项目管理者发现当前所采用的软件过程的缺陷，以便对系统进行升级时进行改进。

项目背景

“南大测试在线管理平台”是一个接受客户在线提交委托，由南京大学软件测试中心分配人员，帮助客户进行软件测试的应用。

“南大测试在线管理平台”为客户与测试中心之间的交互提供平台，客户可以使用移动设备或计算机在网页上进行操作，上传各类文件，并查看自己委托的当前进展；测试中心可以通过平台发送测试文件、测试报告给对应的客户。平台会保留委托进行过程中的每一次操作记录，并生成历史记录以供客户或管理员进行查询。

参考资料

- ruoyi-vue-pro 开发指南
- 测试报告实例

测试概要

测试环境

类型	名称
操作系统	Windows 10，11或Linux
IDE	Intellij IDEA
支撑软件	JDK 11，H2

测试工具

单元测试使用Junit。

Junit是一个开源的Java单元测试框架，是Java的标准单元测试库。Junit测试是程序员主导的测试，即白盒测试，以证明某段代码的行为确实与开发者所期望的一致。Junit的断言机制，可以直接将我们的预期结果和程序运行的结果进行一个比对，确保对结果的可预知性。

性能测试使用JunitPerf,JunitPerf是基于装饰器的Junit扩展框架,测试中通过JunitPerf多线程调用接口测试并可以规定预计时间。

集成测试使用Apifox的测试套件，测试套件为测试用例的集合，每个测试套件包含多个测试用例。

测试内容

测试过程

单元测试

单元测试对各个部分Service进行测试，为了与项目真实数据库隔离开来，单元测试使用H2数据库，每个单元测试运行前会首先运行H2建表语句并在运行结束后进行H2删库操作。

利用Mockito框架虚拟出一个外部依赖，降低测试组件之间的耦合度，只注重代码的流程与结果，真正地实现测试目的。在测试某个Service环节中，单元测试会模拟出其他Service（用@MockBean引入），但是对测试Service和相关数据（Mapper）真实引入（@Resource）。

委托

单元测试列表（委托）

序号	文件名	类名	方法名	描述
1	DelegationServiceImplTest	DelegationServiceImplTest	creatDelegation	测试创建新委托
2	DelegationServiceImplTest	DelegationServiceImplTest	updateDelegation	测试更新文档材料的url或委托的名称
3	DelegationServiceImplTest	DelegationServiceImplTest	submitDelegation	测试客户-保存软件项目委托测试申请表
4	DelegationServiceImplTest	DelegationServiceImplTest	saveDelegationTable2	客户-保存软件项目委托测试申请表
5	DelegationServiceImplTest	DelegationServiceImplTest	saveDelegationTable3	客户-保存委托测试软件功能列表
6	DelegationServiceImplTest	DelegationServiceImplTest	saveDelegationTable14	测试部人员-保存软件文档评审表
7	DelegationServiceImplTest	DelegationServiceImplTest	distributeDelegation2Mkt	测试市场部主管-分配委托给市场部人员
8	DelegationServiceImplTest	DelegationServiceImplTest	distributeDelegation2Test	测试测试部主管-分配委托给测试部人员
9	DelegationServiceImplTest	DelegationServiceImplTest	auditDelegationSuccessMkt	测试市场部人员-审核委托通过
10	DelegationServiceImplTest	DelegationServiceImplTest	auditDelegationSuccessTest	测试测试部人员-审核委托通过
11	DelegationServiceImplTest	DelegationServiceImplTest	auditDelegationFailMkt	市场部人员-审核委托不通过
12	DelegationServiceImplTest	DelegationServiceImplTest	auditDelegationFailTest	测试测试部人员-审核委托不通过
13	DelegationServiceImplTest	DelegationServiceImplTest	saveOffer	测试市场部人员-保存报价单
14	DelegationServiceImplTest	DelegationServiceImplTest	submitOffer	测试市场部人员-提交报价单
15	DelegationServiceImplTest	DelegationServiceImplTest	rejectOffer	测试客户-不接受报价
16	DelegationServiceImplTest	DelegationServiceImplTest	acceptOffer	测试客户-接受报价
17	DelegationServiceImplTest	DelegationServiceImplTest	deleteDelegation	测试根据id删除委托
18	DelegationServiceImplTest	DelegationServiceImplTest	cancelDelegationClient	测试客户-取消委托
19	DelegationServiceImplTest	DelegationServiceImplTest	cancelDelegationAdmin	测试管理员-取消委托

因为单元测试逻辑基本相同,下面就提交委托单元测试的逻辑加以说明。

```java
public void submitDelegation() {
    Mockito.when(userService.getUser(any())).thenReturn(new AdminUserDO());

    long delegationId = UUID.randomUUID().getMostSignificantBits() & Long.MAX_VALUE;

    DelegationDO del = DelegationDO.builder()
            .id(delegationId)
            .state(DelegationStateEnum.DELEGATE_WRITING.getState())
            .table2Id(randomString())
            .table3Id(randomString())
            .launchTime(new Date())
            .name(randomString())
            .build();

    del.setCreateTime(new Date());
    del.setUpdateTime(new Date());
    del.setDeleted(false);

    delegationMapper.insert(del);
    DelegationSubmitReqVO submitReqVO = randomPojo(DelegationSubmitReqVO.class,o->{
        o.setId(delegationId);
    });

    delegationService.submitDelegation(submitReqVO);

    DelegationDO delegationDO = delegationMapper.selectById(delegationId);

    assertEquals(delegationDO.getState(),DelegationStateEnum.WAIT_MARKETING_DEPARTMENT_ASSIGN_STAFF.getState());
}
```

因为DelegationService的submitDelegation方法需要获得相关user的相关身份,这里

```
Mockito.when(userService.getUser(any())).thenReturn(new AdminUserDO());
```

会返回一个用户,下面使用UUID生成一个id是因为之后需要多线程测试,防止多个线程向数据库里多次插入同一个ID。

之后构造一个DelegationDO插入数据库,注意此时需要将其状态改为待提交,因为这里单元测试多是对流程是否正确进行的测试,构造的新数据需要处于待提交的状态,然后构造请求参数submitReqVO调用DelegationService的submitDelegation,最后通过之前生成delegationId到数据库里查询DelegationDO的状态是否是委托提交后应有的状态。

合同

单元测试列表（合同）

序号	文件名	类名	方法名	描述
1	ContractServiceImplTest	ContractServiceImplTest	createContract	测试市场部人员-创建合同
2	ContractServiceImplTest	ContractServiceImplTest	saveContractTable4	测试客户/市场部人员-保存软件委托测试合同
3	ContractServiceImplTest	ContractServiceImplTest	saveContractTable5	测试客户/市场部人员-保存软件项目委托测试保密协议
4	ContractServiceImplTest	ContractServiceImplTest	submitContractStaff	测试市场部人员-提交合同
5	ContractServiceImplTest	ContractServiceImplTest	submitContractClient	测试客户-提交合同草稿
6	ContractServiceImplTest	ContractServiceImplTest	acceptContractClient	测试客户-接受市场部合同草稿
7	ContractServiceImplTest	ContractServiceImplTest	rejectContractClient	测试客户-不接受市场部合同草稿
8	ContractServiceImplTest	ContractServiceImplTest	rejectContractStaff	测试市场部人员-审核合同不通过
9	ContractServiceImplTest	ContractServiceImplTest	acceptContractStaff	测试市场部人员-审核合同通过
10	ContractServiceImplTest	ContractServiceImplTest	uploadDocument	测试市场部人员-上传实体合同材料的url

样品

单元测试列表（样品）				
序号	文件名	类名	方法名	描述
1	SampleServiceImplTest	SampleServiceImplTest	createSample	测试客户-创建样品
2	SampleServiceImplTest	SampleServiceImplTest	updateSample	测试客户-更新样品
3	SampleServiceImplTest	SampleServiceImplTest	submitSample	测试客户-提交样品
4	SampleServiceImplTest	SampleServiceImplTest	auditSampleSuccess	测试市场部/测试部负责人-样品验收通过
5	SampleServiceImplTest	SampleServiceImplTest	auditSampleFail	测试市场部/测试部负责人-样品验收不通过，用户修改中
6	SampleServiceImplTest	SampleServiceImplTest	deleteSample	测试删除样品
7	SampleServiceImplTest	SampleServiceImplTest	getSample	测试获得样品
8	SampleServiceImplTest	SampleServiceImplTest	getSampleList	测试获得样品列表
9	SampleServiceImplTest	SampleServiceImplTest	getSamplePage	测试获得样品分页

测试方法

单元测试列表（测试方法）				
序号	文件名	类名	方法名	描述
1	SolutionServiceImplTest	SolutionServiceImplTest	createSolution	测试测试部人员-创建测试方案
2	SolutionServiceImplTest	SolutionServiceImplTest	saveSolutionTable6	测试测试部人员-保存软件测试方案表格
3	SolutionServiceImplTest	SolutionServiceImplTest	saveSolutionTable13	测试质量部人员-保存测试方案评审表
4	SolutionServiceImplTest	SolutionServiceImplTest	submitSolutionTable6	测试测试部人员-提交软件测试方案表
5	SolutionServiceImplTest	SolutionServiceImplTest	auditSuccess	测试质量部人员-测试方案审核通过
6	SolutionServiceImplTest	SolutionServiceImplTest	auditFail	测试质量部人员-测试方案审核未通过
7	SolutionServiceImplTest	SolutionServiceImplTest	updateSolution	测试更新测试方案
8	SolutionServiceImplTest	SolutionServiceImplTest	deleteSolution	测试删除测试方案
9	SolutionServiceImplTest	SolutionServiceImplTest	getSolution	测试获得测试方案
10	SolutionServiceImplTest	SolutionServiceImplTest	getSolutionList	测试获得测试方案列表
11	SolutionServiceImplTest	SolutionServiceImplTest	getSolutionPage	测试获得测试方案分页

测试报告

单元测试列表（测试报告）

序号	文件名	类名	方法名	描述
1	ReportServiceImplTest	ReportServiceImplTest	createReport	测试测试部人员-创建测试报告
2	ReportServiceImplTest	ReportServiceImplTest	saveReportTable7	测试保存软件测试报告
3	ReportServiceImplTest	ReportServiceImplTest	saveReportTable8	测试保存测试用例（电子记录）
4	ReportServiceImplTest	ReportServiceImplTest	saveReportTable9	测试保存软件测试记录（电子记录）
5	ReportServiceImplTest	ReportServiceImplTest	saveReportTable10	测试保存测试报告检查表
6	ReportServiceImplTest	ReportServiceImplTest	saveReportTable11	测试保存软件测试问题清单（电子记录）
7	ReportServiceImplTest	ReportServiceImplTest	submitReport	测试测试部人员-提交测试报告
8	ReportServiceImplTest	ReportServiceImplTest	acceptReportManager	测试测试部主管-审核测试报告通过
9	ReportServiceImplTest	ReportServiceImplTest	rejectReportManager	测试测试部主管-审核测试报告不通过
10	ReportServiceImplTest	ReportServiceImplTest	acceptReportClient	测试客户-审核测试报告通过
11	ReportServiceImplTest	ReportServiceImplTest	rejectReportClient	测试客户-审核测试报告不通过
12	ReportServiceImplTest	ReportServiceImplTest	acceptReportSignatory	测试授权签字人-审核测试报告通过
13	ReportServiceImplTest	ReportServiceImplTest	rejectReportSignatory	测试授权签字人-审核测试报告不通过
14	ReportServiceImplTest	ReportServiceImplTest	archiveReport	测试测试部-归档测试报告
15	ReportServiceImplTest	ReportServiceImplTest	sendReport	测试市场部-发送测试报告
16	ReportServiceImplTest	ReportServiceImplTest	receiveReport	测试客户-确认接收测试报告
17	ReportServiceImplTest	ReportServiceImplTest	deleteReport	测试删除测试报告
18	ReportServiceImplTest	ReportServiceImplTest	getReport	测试获得测试报告

公司

单元测试列表（公司）

序号	文件名	类名	方法名	描述
1	CompanyServiceImplTest	CompanyServiceImplTest	createCompany	测试创建公司
2	CompanyServiceImplTest	CompanyServiceImplTest	updateCompany	测试更新公司
3	CompanyServiceImplTest	CompanyServiceImplTest	deleteCompany	测试删除公司
4	CompanyServiceImplTest	CompanyServiceImplTest	getCompany	测试获得公司
5	CompanyServiceImplTest	CompanyServiceImplTest	getCompanyList	测试获得公司列表
6	UserCompanyServiceImplTest	UserCompanyServiceImplTest	createUserCompany	测试创建用户公司
7	UserCompanyServiceImplTest	UserCompanyServiceImplTest	updateUserCompany	测试更新用户公司
8	UserCompanyServiceImplTest	UserCompanyServiceImplTest	deleteUserCompany	测试删除用户公司
9	UserCompanyServiceImplTest	UserCompanyServiceImplTest	getUserCompany	测试获得用户公司
10	UserCompanyServiceImplTest	UserCompanyServiceImplTest	createUserCompanyByCode	测试通过code创建用户公司
11	UserCompanyServiceImplTest	UserCompanyServiceImplTest	getCompanyByUser	测试通过用户获得公司
12	UserCompanyServiceImplTest	UserCompanyServiceImplTest	assignNormalUserRole	测试分配普通用户角色

#####

授权

单元测试列表（授权）				
序号	文件名	类名	方法名	描述
1	FrontMenuServiceImplTest	FrontMenuServiceImplTest	createFrontMenu	测试创建权限菜单
2	FrontMenuServiceImplTest	FrontMenuServiceImplTest	updateFrontMenu	测试更新权限菜单
3	FrontMenuServiceImplTest	FrontMenuServiceImplTest	deleteFrontMenu	测试删除权限菜单
4	FrontMenuServiceImplTest	FrontMenuServiceImplTest	getFrontMenu	测试获得权限菜单
5	FrontMenuServiceImplTest	FrontMenuServiceImplTest	getFrontMenuList	测试获得权限菜单列表
6	FrontMenuServiceImplTest	FrontMenuServiceImplTest	getFrontMenuPage	测试获得权限菜单分页
7	FrontMenuServiceImplTest	FrontMenuServiceImplTest	validFrontMenus	测试确认权限菜单
8	FrontPermissionServiceImplTest	FrontPermissionServiceImplTest	getRoleMenuIds	测试获得角色菜单id
9	FrontPermissionServiceImplTest	FrontPermissionServiceImplTest	assignRoleMenu	测试分配角色菜单
10	FrontPermissionServiceImplTest	FrontPermissionServiceImplTest	getUserRoleIdListByUserId	测试通过userid获得角色用户列表
11	FrontPermissionServiceImplTest	FrontPermissionServiceImplTest	assignUserRole	测试分配角色用户
12	FrontPermissionServiceImplTest	FrontPermissionServiceImplTest	getUserRoleIdListByRoleId	测试通过角色id获得用户角色列表
13	FrontPermissionServiceImplTest	FrontPermissionServiceImplTest	getSimpleUserListByRoleId	测试通过角色id获得简单用户列表
14	FrontPermissionServiceImplTest	FrontPermissionServiceImplTest	getRoleMenuList	测试获得角色菜单列表

日志

单元测试列表（日志）				
序号	文件名	类名	方法名	描述
1	FlowLogServiceImplTest	FlowLogServiceImplTest	saveLog	测试保存日志
2	FlowLogServiceImplTest	FlowLogServiceImplTest	listLogs	测试获得日志列表
3	FlowLogServiceImplTest	FlowLogServiceImplTest	createFlowLog	测试创建日志
4	FlowLogServiceImplTest	FlowLogServiceImplTest	updateFlowLog	测试保存日志
5	FlowLogServiceImplTest	FlowLogServiceImplTest	deleteFlowLog	测试删除日志
6	FlowLogServiceImplTest	FlowLogServiceImplTest	getFlowLog	测试获得日志
7	FlowLogServiceImplTest	FlowLogServiceImplTest	getFlowLogList	测试获得日志列表
8	FlowLogServiceImplTest	FlowLogServiceImplTest	getFlowLogPage	测试获得日志分页

性能测试

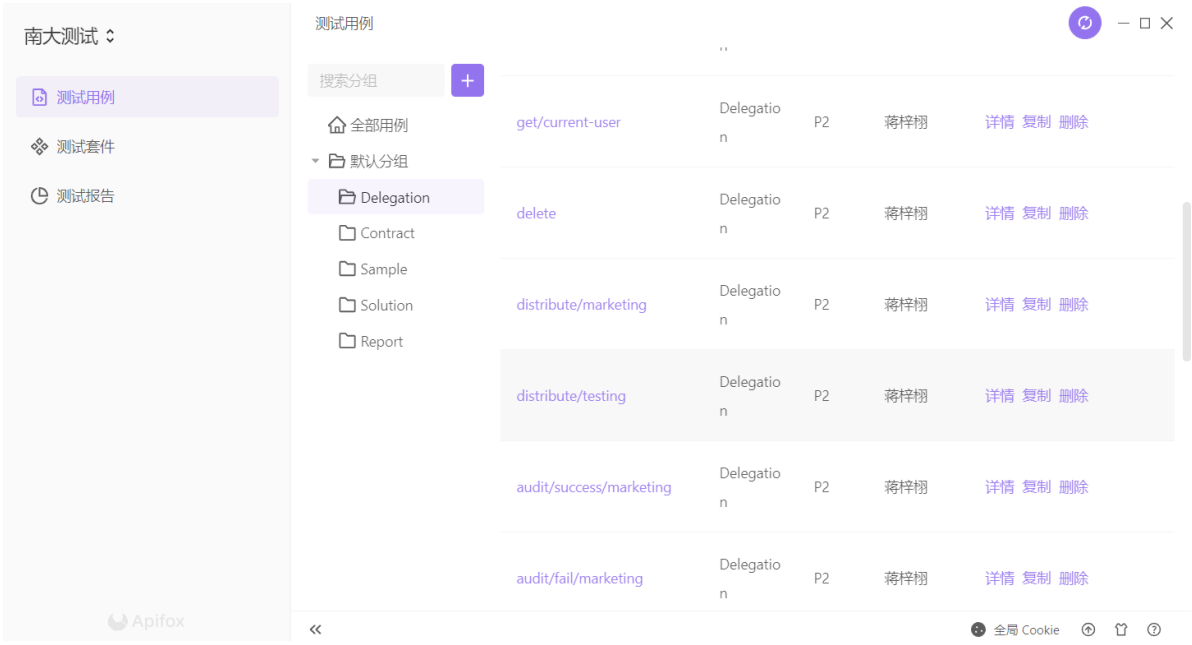
通过JunitPerf在Junit原本的注解上增加如下注解:

```
 @JunitPerfConfig(threads = 8, warmUp = 0, duration = 1000,reporter =
{HtmlReporter.class})
    @JunitPerfRequire(min = 210, max = 250, average = 225, timesPerSecond = 4,
percentiles = {"20:220", "50:230"})
```

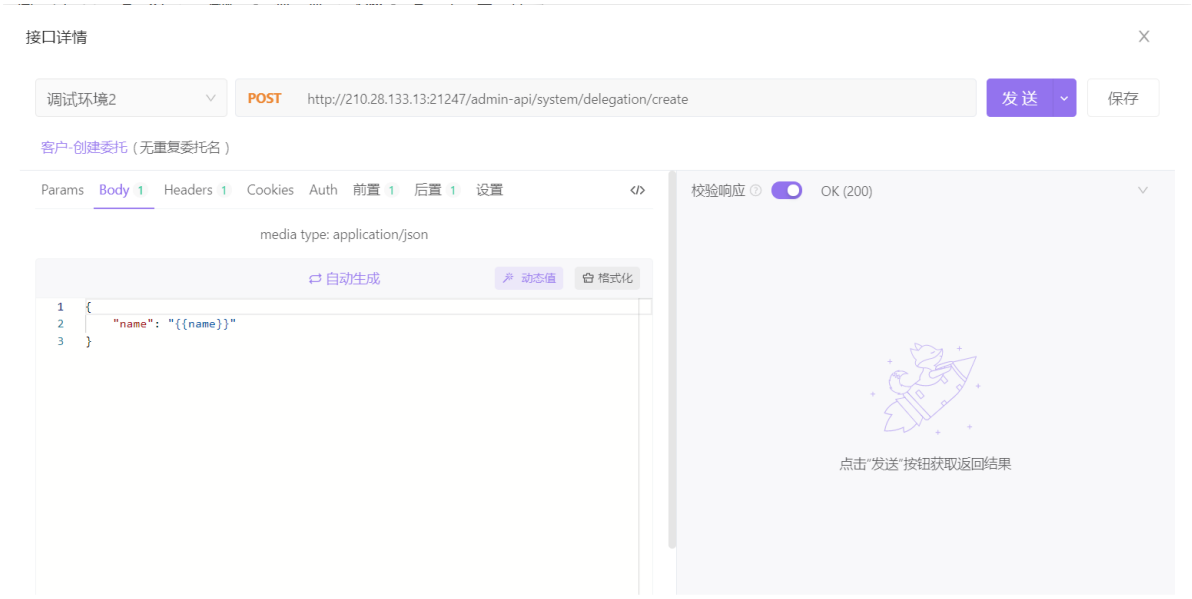
约定同时开启8个线程开始测试,最大时延不得超过250ms,最低时延不得低于210ms,每秒调用不得低于4次。

集成测试

使用Apifox的自动化测试,创建各个阶段测试用例并发出请求:

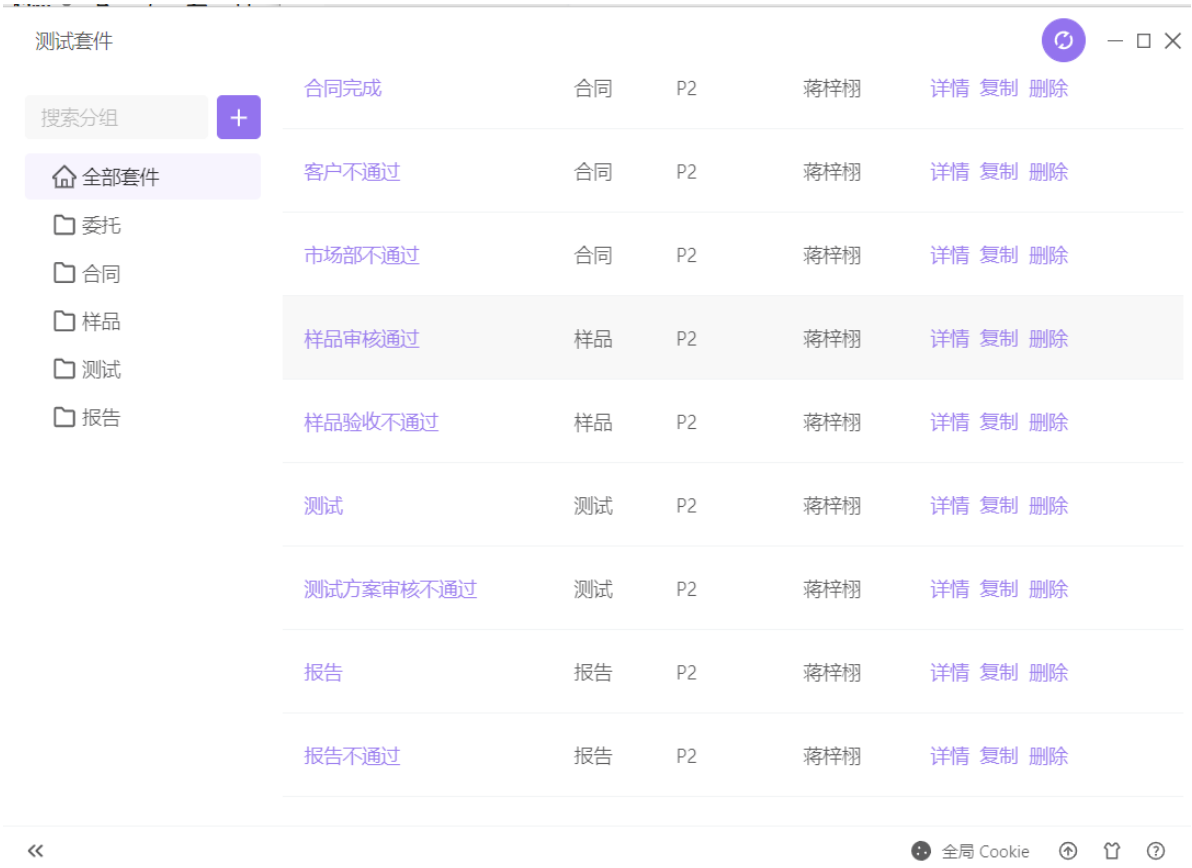


具体测试客户-创建委托如下:



集成测试对过程中需要的变量提取保存并加入适当数据库操作,将测试用例按顺序组成测试套件,

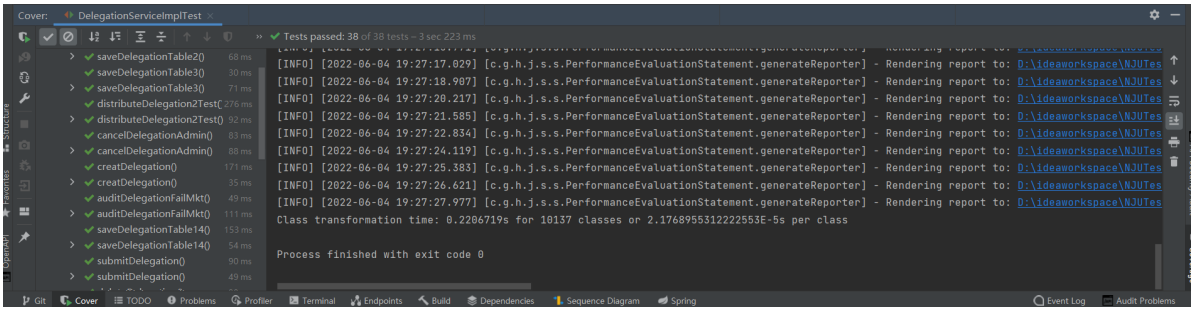
考虑流程图中的不同意打回组成多个测试套件。



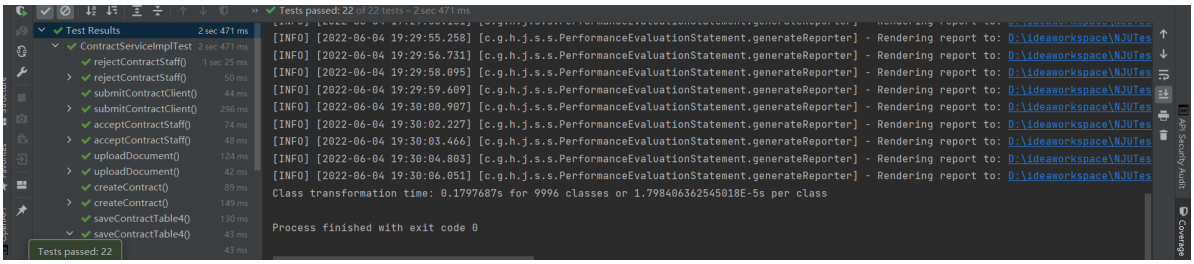
测试结果

单元测试

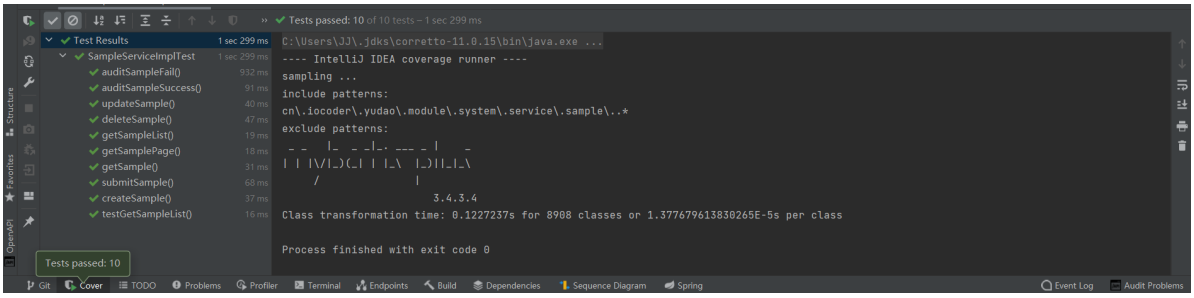
委托



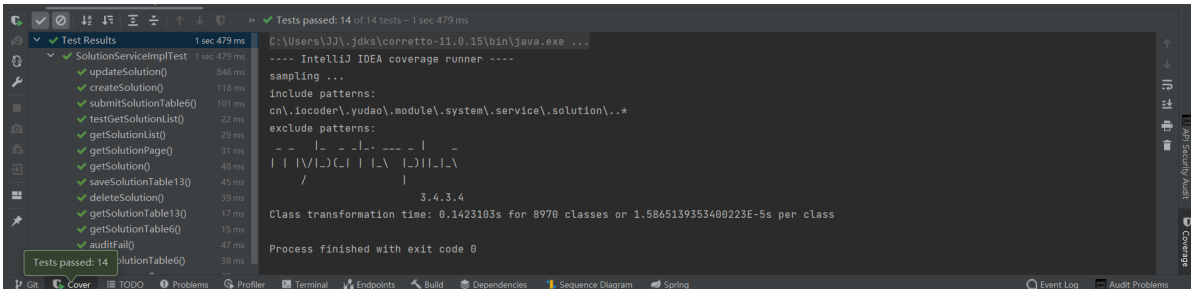
合同



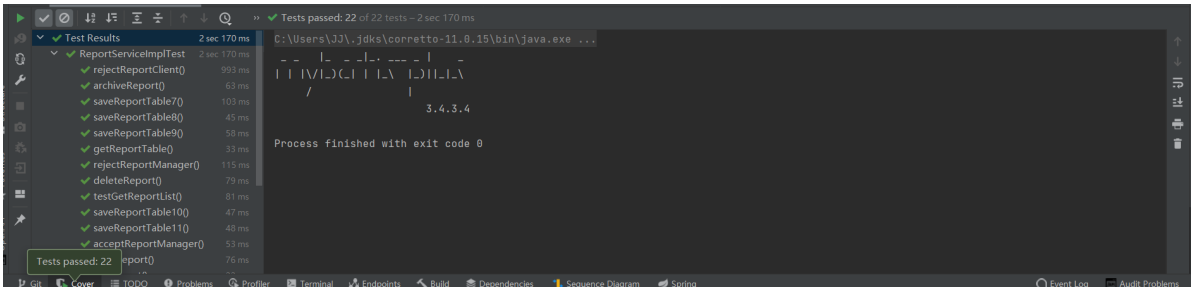
样品



测试方案



测试报告



公司

日志

测试覆盖率如下:

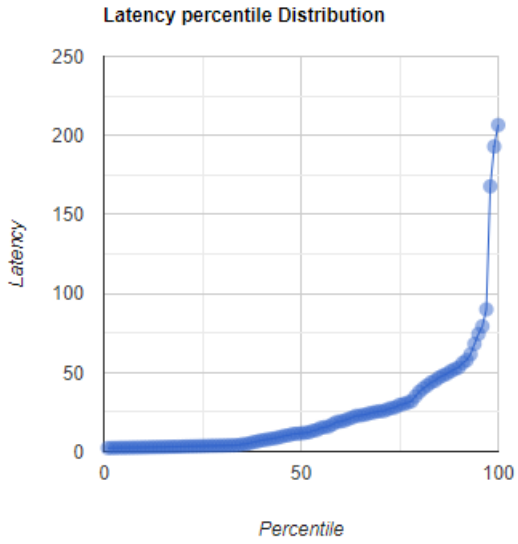
98% classes, 82% lines covered in 'all classes in scope'

Element	Class, %	Method...	Line, %
cn.iocoder.yudao.mod...	100% (2...	74% (20...	72% (62...
cn.iocoder.yudao.mod...	100% (9...	77% (21...	90% (10...
cn.iocoder.yudao.mod...	100% (1...	69% (37...	73% (16...
cn.iocoder.yudao.mod...	100% (1...	100% (1...	90% (27...
cn.iocoder.yudao.mod...	91% (11...	78% (29...	85% (15...
cn.iocoder.yudao.mod...	100% (5...	82% (14...	90% (68...
cn.iocoder.yudao.mod...	100% (5...	70% (14...	85% (66...

性能测试

委托(测试结果以委托为例)

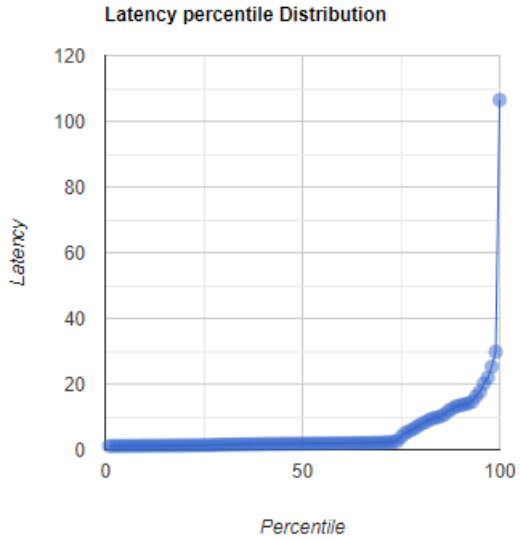
saveDelegationTable2



- Invocations : 343
- Success: 343
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:20.255

Type	Actual	Required
Throughput	343 / s	4 / s
Min latency	2.182 ms	210 ms
Avg latency	23.182 ms	225 ms
Max latency	206.608 ms	250 ms

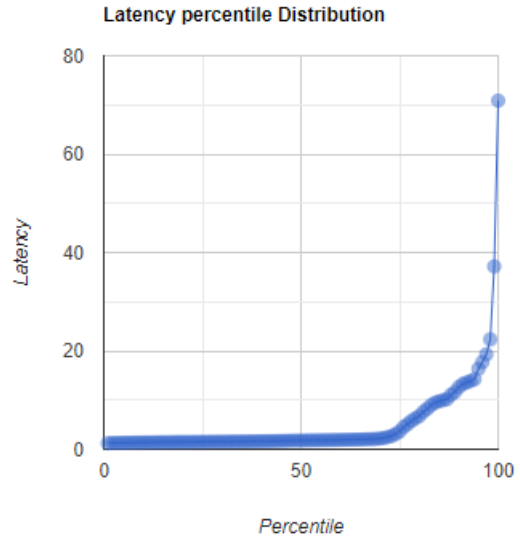
saveDelegationTable3



- Invocations : 1,731
- Success: 1,731
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:21.993

Type	Actual	Required
Throughput	1,723 / s	4 / s
Min latency	1.085 ms	210 ms
Avg latency	4.624 ms	225 ms
Max latency	106.503 ms	250 ms

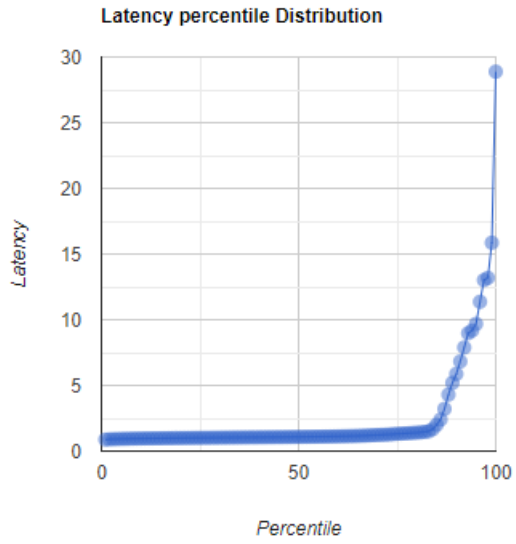
distributeDelegation2Test



- Invocations : 1,778
- Success: 1,770
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:23.179

Type	Actual	Required
Throughput	1,777 / s	4 / s
Min latency	1.201 ms	210 ms
Avg latency	4.541 ms	225 ms
Max latency	70.882 ms	250 ms

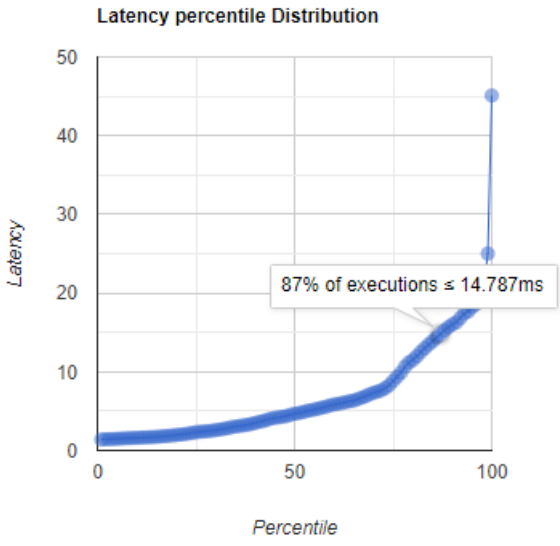
cancelDelegationAdmin



- Invocations : 3,728
- Success: 3,706
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:24.314

Type	Actual	Required
Throughput	3,721 / s	4 / s
Min latency	0.091 ms	210 ms
Avg latency	2.223 ms	225 ms
Max latency	28.884 ms	250 ms

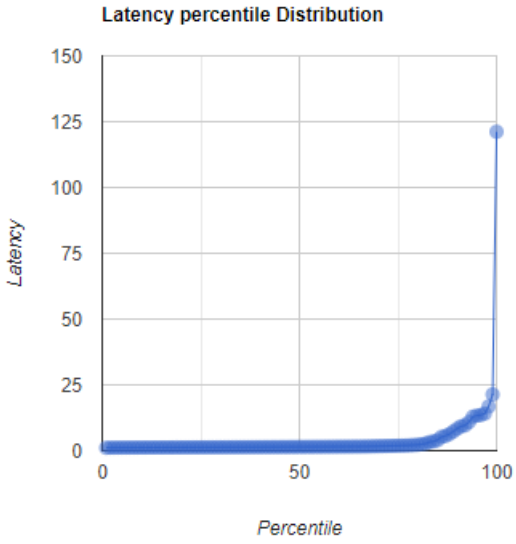
creatDelegation



- Invocations : 1,220
- Success: 1,220
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:25.508

Type	Actual	Required
Throughput	1,212 / s	4 / s
Min latency	1.3 ms	210 ms
Avg latency	6.686 ms	225 ms
Max latency	45.101 ms	250 ms

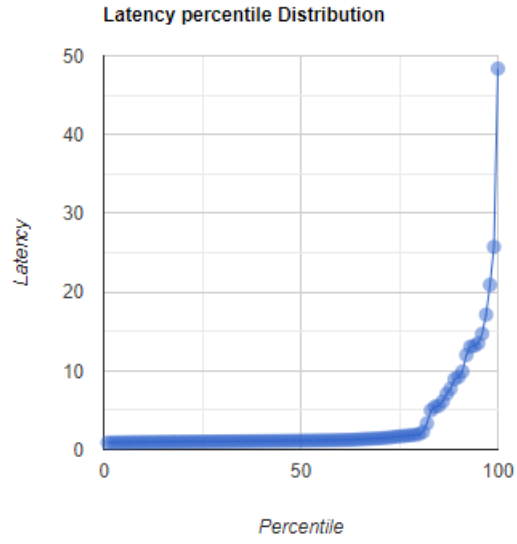
auditDelegationFailMkt



- Invocations : 2,519
- Success: 2,507
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:26.616

Type	Actual	Required
Throughput	2,511 / s	4 / s
Min latency	0.088 ms	210 ms
Avg latency	3.239 ms	225 ms
Max latency	121.18 ms	250 ms

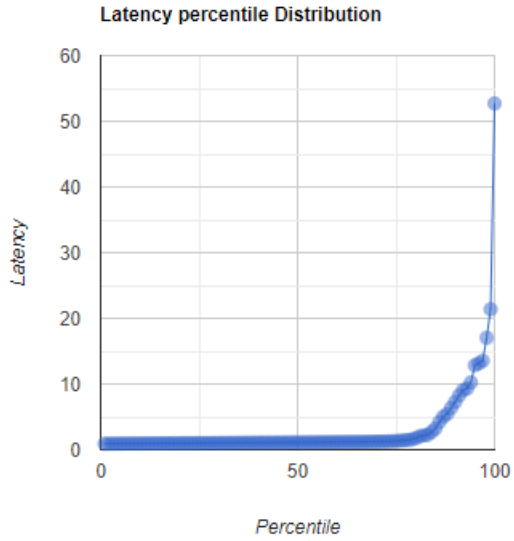
saveDelegationTable14



- Invocations : 2,540
- Success: 2,535
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:27.803

Type	Actual	Required
Throughput	2,534 / s	4 / s
Min latency	0.195 ms	210 ms
Avg latency	3.131 ms	225 ms
Max latency	48.377 ms	250 ms

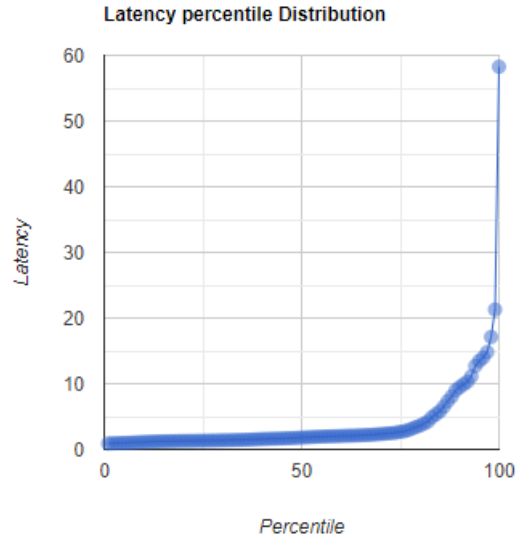
submitDelegation



- Invocations : 3,029
- Success: 3,016
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:28.937

Type	Actual	Required
Throughput	3,024 / s	4 / s
Min latency	0.085 ms	210 ms
Avg latency	2.598 ms	225 ms
Max latency	52.712 ms	250 ms

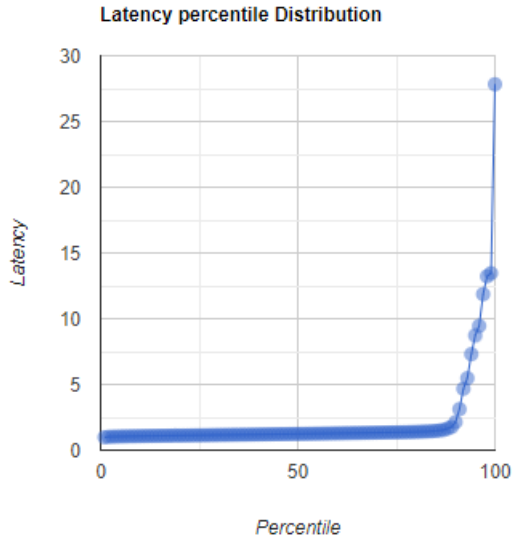
deleteDelegation



- Invocations : 2,357
- Success: 228
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:30.097

Type	Actual	Required
Throughput	2,349 / s	4 / s
Min latency	0.172 ms	210 ms
Avg latency	3.46 ms	225 ms
Max latency	58.268 ms	250 ms

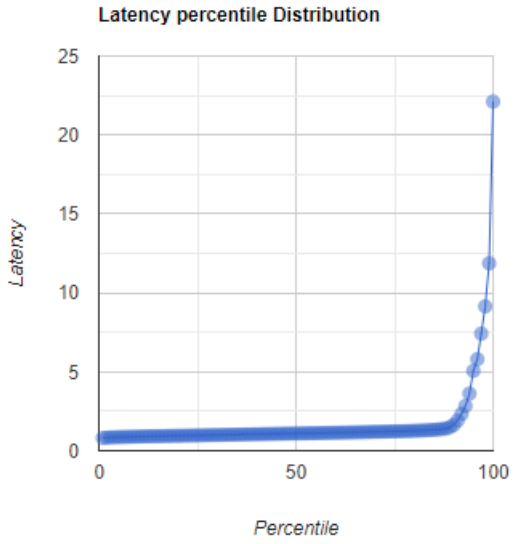
auditDelegationFailTest



- Invocations : 4,033
- Success: 3,982
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:31.265

Type	Actual	Required
Throughput	4,027 / s	4 / s
Min latency	0.066 ms	210 ms
Avg latency	1.986 ms	225 ms
Max latency	27.832 ms	250 ms

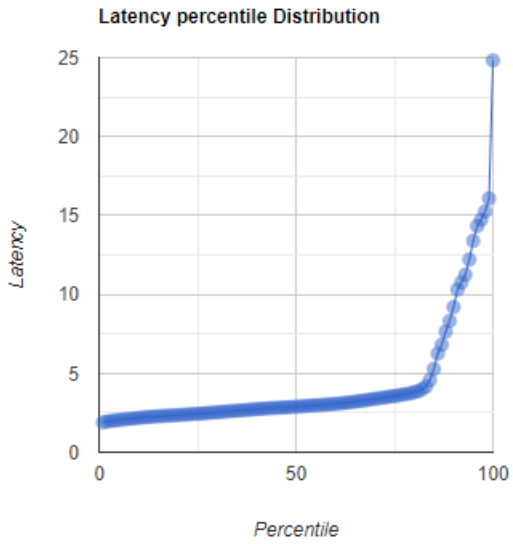
submitOffer



- Invocations : 5,166
- Success: 5,108
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:32.383

Type	Actual	Required
Throughput	5,158 / s	4 / s
Min latency	0.096 ms	210 ms
Avg latency	1.551 ms	225 ms
Max latency	22.116 ms	250 ms

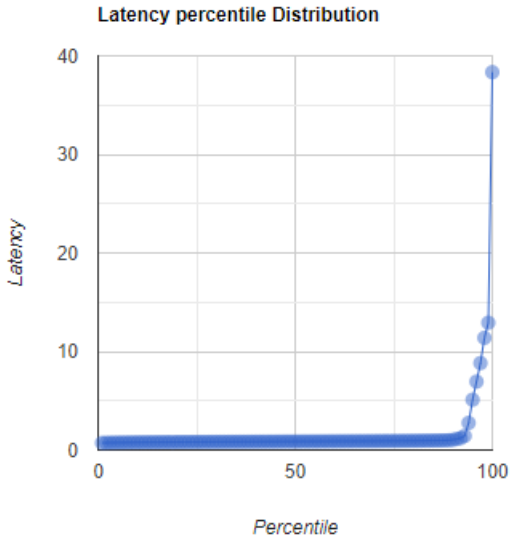
updateDelegation



- Invocations : 1,956
- Success: 1,941
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:33.515

Type	Actual	Required
Throughput	1,948 / s	4 / s
Min latency	0.273 ms	210 ms
Avg latency	4.099 ms	225 ms
Max latency	24.824 ms	250 ms

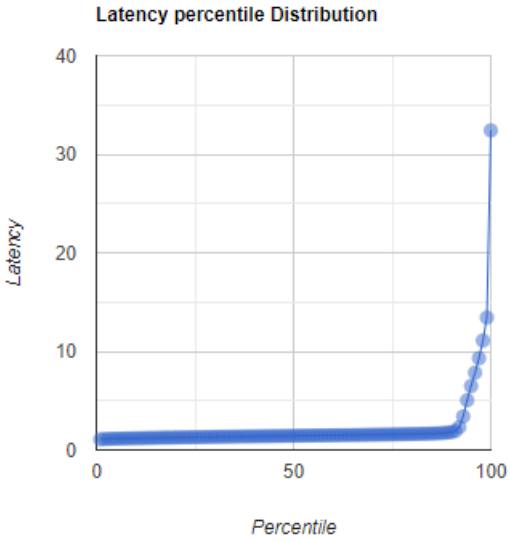
saveOffer



- Invocations : 5,795
- Success: 5,745
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:34.642

Type	Actual	Required
Throughput	5,787 / s	4 / s
Min latency	0.118 ms	210 ms
Avg latency	1.38 ms	225 ms
Max latency	38.343 ms	250 ms

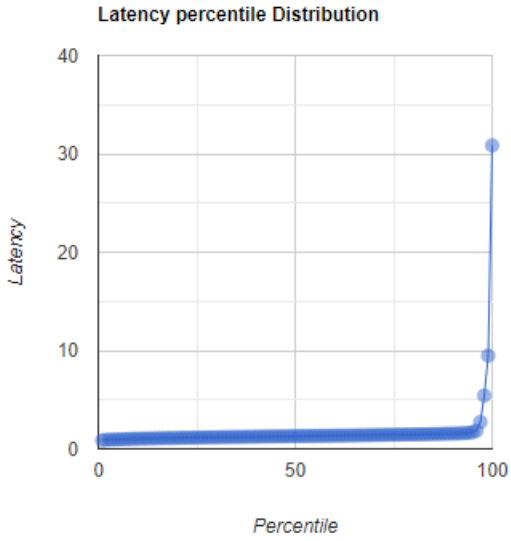
auditDelegationSuccessTest



- Invocations : 4,126
- Success: 4,092
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:35.788

Type	Actual	Required
Throughput	4,118 / s	4 / s
Min latency	0.084 ms	210 ms
Avg latency	1.986 ms	225 ms
Max latency	32.417 ms	250 ms

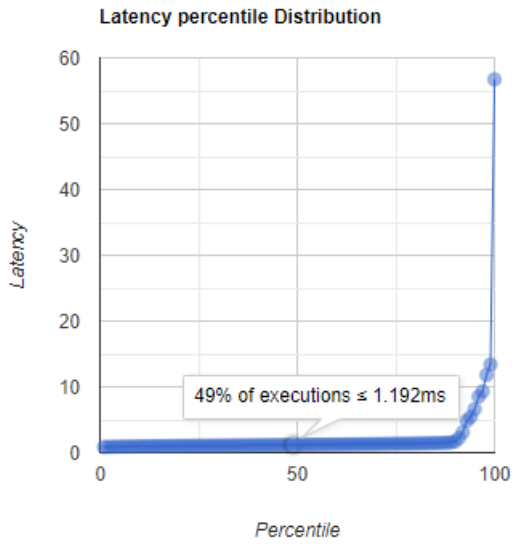
auditDelegationSuccessMkt



- Invocations : 5,447
- Success: 5,363
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:36.972

Type	Actual	Required
Throughput	5,440 / s	4 / s
Min latency	0.099 ms	210 ms
Avg latency	1.516 ms	225 ms
Max latency	30.859 ms	250 ms

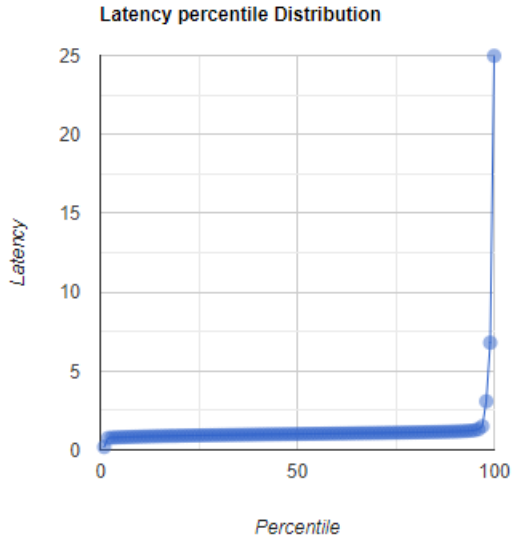
acceptOffer



- Invocations : 4,319
- Success: 4,267
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:38.177

Type	Actual	Required
Throughput	4,312 / s	4 / s
Min latency	0.07 ms	210 ms
Avg latency	1.867 ms	225 ms
Max latency	56.742 ms	250 ms

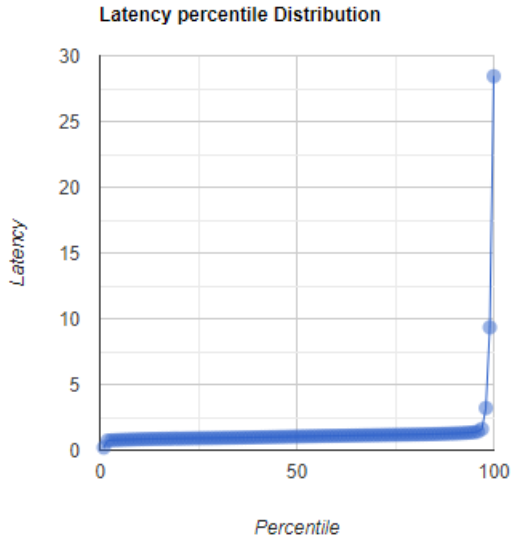
rejectOffer



- Invocations : 6,997
- Success: 6,831
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:39.359

Type	Actual	Required
Throughput	6,991 / s	4 / s
Min latency	0.067 ms	210 ms
Avg latency	1.165 ms	225 ms
Max latency	24.977 ms	250 ms

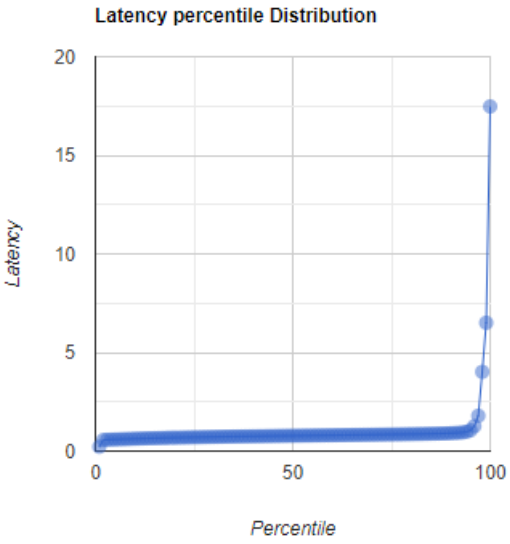
distributeDelegation2Mkt



- Invocations : 6,520
- Success: 6,261
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:40.585

Type	Actual	Required
Throughput	6,512 / s	4 / s
Min latency	0.053 ms	210 ms
Avg latency	1.233 ms	225 ms
Max latency	28.443 ms	250 ms

cancelDelegationClient



- Invocations : 8,816
- Success: 8,673
- Thread Count : 8
- Warm up : 0 ms
- Execution Time : 1,000 ms
- Memory: 32 byte
- Started at: 2022-06-05 06:04:41.798

Type	Actual	Required
Throughput	8,809 / s	4 / s
Min latency	0.05 ms	210 ms
Avg latency	0.922 ms	225 ms
Max latency	17.477 ms	250 ms

集成测试

Apifox 报告

测试用例/测试套件	报告不通过
运行时间	2022-05-29 21:27:25
运行工具	Apifox v2.1.15

	总次数	失败数
循环	51	0
请求	51	0
断言	0	0

总耗时	7.7s
总返回数据量 (约等于)	1.54KB
平均接口请求耗时	152ms

总失败 **0**

测试总结

测试已完成部分，功能正确，运行良好。

数据库中某些数据设置长度过少导致插入失败，现已修改。