OKRs Planning & Review

	Period	OKR X 「Planning Date」「Review Date」					
	Company	Team/Group	Key Results	Key Results	Key Results	Key Results	Key Results
No.	Objectives	Objectives	1	2	3	4	5
1		MAX					
		替代更多业务计算	MAX「计算流程」，拼接测试成功100%（20次/20次）「没有BI」	并发至少要双（2）用户计算操作	账号体系可以达到不同的权限可以进入不同的产品序列	Git仓库以阿斯泰来&辉瑞为参考，成功	Python完成，可以回归到「公司对数」中，缺少Debug的1人来加数
3		TMIST 培训 「APM」					
		获得XU老师认可	R计算逻辑确认，并调通，可以被Call	产品中的「产品」、「场景」、「医院/区域」、「代表」和「定性」及「定量」的值和描述被确认	账号体系可以达到不同的权限可以进入不同的产品序列		
		应用于教学场景		功能性测试、回归测试，完成「测试用例」并确定测试人员，进行测试		压力测试：50人，同时访问保证成功	
2		TMIST 测评					
		支持客户演示	产品完成清晰的页面逻辑，为下1次页面逻辑迭代做好准备		账号体系可以达到不同的权限可以进入不同的产品序列		
			Q 技术债 是否能够切入到并行APM，以及能够快速形成1套样子货，将可能涉及到通信协议				
4		BI					
		能够应用在「礼来项目」之外进行展示					
5		官网					
		能够支持更准确的销售线索获取					

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. 自学操作系统原理（CPU，线程，内存）		
2. 准备UCB		
3. Akka学习（真的看不懂，思想难掌握）		
4. 框架上调		
5. 自动化部署没实现		

Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. Golang的学习势在必行	
3. 设计模式的实际应用		
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	域名、反向代理的配置	30%	为了更好的提高用户体验，避免IP+Port的项目访问方式，使用阿里云的DNS解析和隐性URL技术，可以实现子域名访问项目	因为阿里云的限制，隐性URL的配置限制只有2条，也就是同一域名下，子域名映射的项目最多只有两个，因此要考虑使用其他方式。	下一步	
Tuesday周二	域名、反向代理的配置	70%	按照杨总的方案，尝试使用Nginx的反向代理技术，不仅可以使用Https提高安全性，还可以对所有的访问（用户访问和接口调用）进行一次安全检查和过滤，实现统一管理。	Nginx的配置不支持IF ELSE,也不支持AND OR，写一些逻辑好丑。	下一步	
Wednesday周三	域名、反向代理的配置	完成	重启MongoDB节点，需要重新配置分片关系，如果primary节点不是原来的节点，程序调用没有问题，但GUI使用会报错【not master】，敬请注意。	完成：目前项目部署情况 官网：pharbers.com www.pharbers.com TM: tm.pharbers.com APM: apm.pharbers.com 教师端：teacher.pharbers.com	归档	
Thursday周四	配合小陆试用线上Max	辉瑞和安斯泰来完成	通过今天对小陆的采访，他还是很满意的，对于放大速度也认可。 但是还是有可以提升的地方： 1. 现在最浪费的时间就是每月更新的产品匹配表，小陆手动修改需要3h左右。 2. 试用线上放大后的结果，需要通过工程团队聚合和下载才行。		归档	
Friday周五	准备OKR Review		操作系统博客：https://www.cnblogs.com/clockq/p/10318639.html OKR_Review PPT: https://clockq.github.io/Pharbers_OKR_Review			
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. 自学操作系统原理（CPU，线程，内存）		
2. 准备UCB		
3. Akka学习（真的看不懂，思想难掌握）		
4. 框架上调		
5. 自动化部署没实现		

Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. Golang的学习势在必行	
3. 设计模式的实际应用		
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	Max 本地集群部署	50%	集群部署问题还是很难估计的。首先YARN部署Spark应用，需要的一些资源文件如何加载？ip能ping通但无法启动Spark？发布应用的依赖要整体打包。XMPP部署和账号问题。	基本已经解决了。	下一步	
Tuesday周二	Max 本地集群部署	100%	实际部署后的日志处理和异常处理有些麻烦。		归档	
Wednesday周三	APM 些微修改	100%	安琪提出的小问题，已经解决。		归档	
Thursday周四	OS 自学	博客ing	敬请期待！！！		归档	
Friday周五	OS 自学	博客ing	敬请期待！！！		归档	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. 自学操作系统原理（CPU，线程，内存）		
2. 准备UCB		
3. Akka学习（真的看不懂，思想难掌握）		
4. 框架上调		
5. 自动化部署没实现		

Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. Golang的学习势在必行	
3. 设计模式的实际应用		
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	Max添加新公司	正大天晴添加完成	算法有些微改变		下一步	
Tuesday周二	Max添加新公司	齐鲁、BMS、倍特添加完成	代码添加完成		归档	
Wednesday周三	新增公司对数		都是之前遇到的问题，难度不大，但耗费时间		下一步	
Thursday周四	新增公司对数		问题和前一天一样，人工干预的工作很难避免出现错误		下一步	
Friday周五	新增公司对数		即将完成		归档	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. 自学操作系统原理（CPU，线程，内存）		
2. 准备UCB		
3. Akka学习（真的看不懂，思想难掌握）		
4. 框架上调		
5. 自动化部署没实现		

Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. Golang的学习势在必行	
3. 设计模式的实际应用		
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	Driver 2 Logic 对接工作	正在和老铁对接	上周基本解决，没有问题		归档	
Tuesday周二	Logic 2 Web 对接工作	主要是老铁和森哥的工作	无		归档	
Wednesday周三	Max功能测试	大家伙一起搞测试	无		下一步	
Thursday周四	Max功能测试	大家伙一起搞测试	无		下一步	
Friday周五	Max功能测试	大家伙一起搞测试	无		归档	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. Max Builder 可配置化（完成）	6. Akka学习（真的看不懂，思想难掌握）	
2. 准备UCB		
3. 代码Review，现在无法AB Test（搁置）		
4. 框架上调		
5. 自动化部署没实现		

Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. 自学操作系统原理（CPU，线程，内存）	
3. 设计模式的实际应用	8. Golang的学习势在必行	
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	1. Max Builder 可配置化	恩华，安斯泰来完成	今天max基本完成，除去前面的大量准备工作，实际开发时间也就今天一天，目前剩下的主要问题就是在多用户执行情况下，yarn只有一个appid，并且进度条错乱，进度条问题有个简单的办法解决，但治标不治本，原因还是进度条所监控的app未分开，还有一个问题就是同一用户或统一jobid重复启动spark的异常处理		下一步	
Tuesday周二	1. Max Builder 可配置化	拜耳完成	基本完成了Driver的编写，开始和老铁联调，因为将Driver改写成了一个算能，主要的执行流程完全通过老铁传过来的协议控制，减少了我的大量工作，但是增加了我俩之间的大量沟通成本，同时因为老铁已经和森哥联调好了，而他们之间通讯的实体类和我有很大出入，导致我们浪费了大量时间。而关于两种协议要不要合并的问题，我和杨总产生了强烈分歧，回去后考虑一下。 还有一个问题就是Go端对JsonApi的转换默认值问题，和我的Scala大相径庭，需要处理很多的默认值。		下一步	
Wednesday周三	1. Max Builder 可配置化	施维雅，天晴，信立泰完成	施维雅测试未通过，要去参加轰趴了，今天进展不大		下一步	
Thursday周四	1. Max Builder 可配置化	辉瑞完成	七个公司都完成了，考虑一下需要优化的地方，以及样本检查，结果检查的接口。 拿下载举例，前端发送请求，go响应并xmpp到scala，同时前端接受响应成功开始loding，scala的spark开始下载，结果存到hdfs，xmpp到前端或go，返回文件路径，正式下载开始。 之后就有一个问题，当go，或者driver进行分布式，xmpp需要使用群组接受，怎么防止重复消费。	防止重复消费，我想到了两套方案。 1. 编码层次，tag锁加全局单例，在群组中，保证一条消息只有一次消费。 2. 架构层次，在Driver前的XMPP加一层反向代理，Go每次只发送给一个driver。	等待测试	
Friday周五	1. Max Builder 可配置化	Max Result 导出（分片聚合）功能完成			等待测试	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. Max Builder 可配置化	6. Akka学习（真的看不懂，思想难掌握）	
2. 准备UCB		
3. 代码Review，现在无法AB Test（搁置）		
4. 框架上调		
5. 自动化部署没实现		

Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. 自学操作系统原理（CPU，线程，内存）	
3. 设计模式的实际应用	8. Golang的学习势在必行	
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	1. Max Builder 可配置化	Nhwa panel 调试完成	进度条修好了，代码改动太大，导致修复的比较缓慢	烦	下一步	
Tuesday周二	1. Max Builder 可配置化	Nhwa 计算也通了，现在可以考虑写Builder了	Astl的计算月份，努力中	很烦	下一步	
Wednesday周三						
Thursday周四						
Friday周五						
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. Max Builder 可配置化	6. Akka学习（真的看不懂，思想难掌握）	
2. 准备UCB		
3. 代码Review，现在无法AB Test（搁置）		
4. 框架上调		
5. 自动化部署没实现		

Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. 自学操作系统原理（CPU，线程，内存）	
3. 设计模式的实际应用	8. Golang的学习势在必行	
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	病倒了					
Tuesday周二	病倒了					
Wednesday周三	1. Max Builder 可配置化	不好描述	果然，每次看自己的老代码，都会有“惊喜”	http://www.uml.org.cn/sjms/201211023.asp 设计模式六项基本原则，记得按照这个优化	下一步	
Thursday周四	1. Max Builder 可配置化	不好描述	XMPP调通了，杨总的代码太抽象了。 单例 + 监听 + AKKA + pool，真的不敢动这代码。 同时发现Max的一些逻辑问题，视情况改动吧。	今天更加理解杨总的老话“组合优于继承”，“实例拥有共同父类（面向接口编程）”，同时，学会了Actor一个很不要脸的操作，只保留一个Helper的Actor句柄，在使用的地方ActorSelection来发送，可以实现和大程度的解耦，太骚了。	下一步	
Friday周五	1. Max Builder 可配置化	不好描述	XMPP果然还是得搞一搞，AKKA逃不了了。 Spark的新集群，在小崔的帮助下，很快配置完成。HDFS上次我怎么都没搞定， 小崔NB。 计算月份和Panel已经把参数提取出来了，并且已经调通了，但是进度条不好用。明天搞。	Akka不会, 被杨总diss了，多线程编程的学习，刻不容缓啊。	下一步	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. APM压测达标（完成）		
2. 权限账号体系要加速建设（完成）		
3. 代码Review，现在无法AB Test		
4. 框架上调		
5. 自动化部署没实现		

Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. 自学操作系统原理（CPU，线程，内存）	
3. 设计模式的实际应用	8. Golang的学习势在必行	
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	1. 完成queryCount 2. 实现王同学的日历小点点（仅获得有提交的日期）	100% 100%	编码问题，真的是跪了，永远都是开发环节的滑铁卢！		归档	
Tuesday周二	1. 重新发布全部节点	40%	为了更加接近五十人目标，去掉很多的不必要工作，包括查询，组合和解析。并且部署的代码也使用主机名进行forward而不是之前的ip加外部端口，可是压测结果较之前还略有下降，问题正在排查。		下一步	
Wednesday周三	1. 重新发布全部节点 2. 调整账号权限系统 3. 1.2版本完成，备份数据库	100% 100% 100%		在今天的修改账号权限过程中，对编程有了一种新的感悟，MVC虽然是很老的思想了，但就像冯诺伊曼体系结构一样影响深远。这不止是程序编写的一种思想，更是计算机领域一切问题的切入点。如果我能在设计权限架构之初，就明确区分数据结构，程序逻辑和老郭的业务逻辑，好像MVC这样的分层，就可以避免大部分问题，我的数据结构抽象的够好，MVC足够独立，就可以让郭哥的V任意改变和组合。	归档	
Thursday周四	1. 对于问题接口就要特殊对待	100%		加入了一层缓存，现在50并发没问题，Redis真是好用啊。 但是要注意，对于只读缓存的脏数据处理很麻烦。要重点研究一下。		
Friday周五						
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务
帮助大家应用「GTD」进行时间管理
请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代		
1. APM压测达标		
2. 权限账号体系要加速建设		
3. 代码Review，现在无法AB Test		
4. 框架上调		
5. 自动化部署没实现		
Your Inbox => 幸存区		
1. 十万并发下的兔子问题(斐波那契数列)怎么做	6. 自学数据结构	
2. 阴险的约瑟夫	7. 自学操作系统原理（CPU，线程，内存）	
3. 设计模式的实际应用	8. Golang的学习势在必行	
4. MongoDB的读写分离		
5. 水位监控实现		

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	1. APM教师端下载编码修改，默认GB2312 2. OKR Review	100% 100%	编码问题，真的是跪了，永远都是开发环节的滑铁卢！		归档	
Tuesday周二	1. 优化APM代码，提高执行效率	40%	分段记录接口执行时间，对于简单接口，查询时间如下： 1. request = 12.5 — 15 2. parseToken = 8 — 10 3. formJsonapi = 1 4. queryMulti = 7 — 9 5. forwardTo = 88 — 93 6. toJsonapi = 32 — 37 7. response = 12.5 — 15 其中， client的响应时间是212ms，除去网络消耗的代码执行时间是192ms，和目前一些线上的10ms内返回比， 还是差太太太多了。	为了比较，找了一个至今最令我头疼的接口（因为返回的数据量太大，相对而言）测试，改了一些东西，如parseToken，测试时间如下： 1. request = 12.5 — 15 2. parseToken = 5 3. formJsonapi = 1 4. queryBindMulti = 19 5. queryMulti = 17 6. toJsonapi = 325 7. response = 12.5 — 15 其中， client的响应时间是436ms，除去网络消耗的代码执行时间是403ms，可以看出，性能瓶颈在toJsonapi这个方法。	下一步	
Wednesday周三	1. 优化APM代码，提高执行效率 2. 配合森哥修复APM的Bug					
Thursday周四	1. 优化APM代码，提高执行效率 2. 配合森哥修复APM的Bug		JS的正则表达式要以”/“开头，以“/”结尾。			
Friday周五	1. 优化APM代码，提高执行效率 2. 配合森哥修复APM的Bug		杨总说的对“软件没有系统边界”，我得好好想想。			
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代

1. MongoDB的读写分离
2. 代码效率优化
3. UML之ER图的学习

Your Inbox => 幸存区

1. 十万并发下的兔子问题(斐波那契数列)怎么做
2. 阴险的约瑟夫
3. 设计模式的实际应用

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	1. APM新需求，可下载学员输入	40%	目前还没有实际需求，所以只能先把通用的方法先设计一下，实际需求不能自己臆想。		下一步	
Tuesday周二	1. APM新需求，可下载学员输入 2. 和郭哥安琪讨论教师流程 3. APM1.1 版本结束，备份全部资料，进入1.2	40%	目前还没有实际需求，所以只能先把通用的方法先设计一下，实际需求就不自己臆想了。		下一步	
Wednesday周三	1. 完成APM的教师端接口 2. 严格执行开发流程，学习drow.io的ERD绘制	50% 等待对接	ERD的一些细节已经忘记了，还要好好学习一下啊		等待对接	
Thursday周四	1. 完成APM的教师端接口	80% 等待测试			等待测试	
Friday周五	1. 完成APM的教师端接口	80% 等待测试			等待测试	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

OKR Review 总结：
收货：
项目：
1. Redis Bug修复，修复过程https://www.cnblogs.com/clockq/p/9946513.html
2. APM替换R的计算
3. RSA和DES加密以及编码问题
4. 法伯的Shiro架构，博客地址https://www.cnblogs.com/clockq/p/9908742.html
5. 解决Sales表遗留的坑
6. Nginx加入
7. Https加入
8. APM教师端加入：

项目：
1. 一周自学Linux的视频，博客地址https://www.cnblogs.com/clockq/p/10007493.html

不足：
项目：
1. APM压测仍未达标
2. MongoDB读写分离还未实现
3. 权限账号体系要加速建设
4. 下载文件编码（真的很烦）
5. 代码Review，现在无法AB Test
6. 框架上调
7. 自动化部署没实现

个人：
1. DP未学未用
2. 水位监控实现

感悟：
1. ERD先行，感悟于（收货 =》 项目 =》5. 解决Sales表遗留的坑）
2. 深入理解微服务（不修改，只增加）

学习与培训纪要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务
帮助大家应用「GTD」进行时间管理
请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」
Your Inbox => 新生代
Your Inbox => 幸存区
1. 十万并发下的兔子问题(斐波那契数列)怎么做
2. 阴险的约瑟夫
然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	为APM上线，学习网站的并发优化技术	**写一个跳转节点的连接池**	连接池其实是关键，要设计完善的话，还是有很多细节要认真考虑： a) 连接池的初始连接数、最低连接数、最大连接数，这些都是要考虑的。 b) 如果长时间没有client调用，要有额外机制来释放闲置连接，可以额外开一个线程定时检测 c) 如果server端，某个实例down掉，要考虑将对应的连接置为不可用，或者直接释放 d) 没有可用连接时，如果池中的连接数<最大连接数，要考虑主动创建新连接。		下一步	
Tuesday周二	为APM上线，应用网站的并发优化技术	学习负载均衡的具体实现	实现负载均衡， 要实现的东西还是不少的，首先需要提供一个注册中心作为外部总控。注册方式又分为主动注册和第三方注册。发现模式有客户端发现模式和服务端发现模式。其次还要考虑错误处理，熵减，负载算法，初步打算使用Akka处理，因为Akka有自己成熟的监管策略和并发效率， 使用Akka写一个单例的计数器，均匀的将请求分配到各个节点。		下一步	
Wednesday周三	增加机器吗？不是，这叫**Docker Swarm Replicas**	100%		“加了机器”果然可以解决产品运维过程中80%的问题，但硬资源有了， 软实力同样重要，继续优化APM。	归档	
Thursday周四	加入Nginx取代Docker的负载均衡	100%	Nginx修改配置无效	需要执行“docker exec -it nginx容器ID nginx -s reload”, 重启nginx	归档	
Friday周五	对于后端接口开启HTTPS	100%	什么是HTTPS？什么是SSL? 登录的阿里云的后台，目录好多， 眼花缭乱。		归档	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查
如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先
最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排
思考/自我评价
学习与培训纪要
Docker Swarm 学习博客： 1. 基于 Docker 的微服务架构实践 2. Docker Swarm集群初探 3. 基于go的微服务搭建（七） - 服务发现和负载均衡 4. 使用 Docker Swarm 管理 Docker 集群

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代

Your Inbox => 幸存区
1. 十万并发下的兔子问题(斐波那契数列)怎么做
2. 阴险的约瑟夫

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	1. 准备PH权限的培训博客 2. 新增APM的显示数据(包括新的APM计算)	100% 20%	关于微服务架构在工程开发过程中, 如何保证稳定版本的同时, 可以适应不断修改的需求(最麻烦的是修改底层数据结构, 导致不向前兼容)		归档 下一步	博客地址: Pharbers用于单点登录的权限架构
Tuesday周二	新增APM的显示数据(包括新的APM计算)	80%	在开发过程中, 还会出现需求变化, 是不合理的.	软件开发的流程是: 1. 需求分析 2.概要设计 3. 详细设计 4. 编码开发 5.测试 6. 发布 如同昨天杨总所讲那样, 我们虽然不严格, 但也应该按照这个流程去执行, 在设计阶段, 也应该绘图先行(ER图, 流程图, 数据流图, 时序图, 泳道图, 这些都要捡起来)	下一步	
Wednesday周三	协助森哥处理bug	100%	修复无数小bug，真的是无数	还是昨天的问题，作为一个程序员，写出一个可以适应，甚至自适应未来需求改动的程序，真的是一门学问。	等待郭哥和安琪测试	
Thursday周四	1. APM的压力测试	1. 90%	并发用户数和并发请求可不是一个东西， 测试真的是一门大学科， 里面学问深着呢。	今天看了一篇网站压测的博客， （28定律）真的是在IT圈无处不在。要培养自己对曲线， 对数据的敏感性。	个别接口还需要重新测试一下， 还有，研究一下完整的并发优化方案	
Friday周五	1. APM的压力测试 2. 学习网站的并发优化技术	1. 100%	如何使用现有资源提高系统并发量? 加硬件资源解决 和 使用技术手段解决的优劣和取舍?		归档	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

回顾最近两周的Scrum， 总结为以下两块：
1. 项目：
  完成情况：
    1.1 APM成功替换Scala的计算
    1.2 类Shiro的登录和注册流程完成
    1.3 Sales数据库的优化
  不足：
    1.4 APM压测不满足徐老师的50人并发（可通过扩展个别节点解决）
    1.5 登录注册流程还可以进一步细化（视具体业务流程而定）
2. 个人：
  收获：
    2.1 Shiro的培训博客：Pharbers用于单点登录的权限架构
    2.2 Docker 的 Redis 安全问题博客：应用篇 = Docker下的Redis
  迷惑：
    2.3 关于微服务下业务的修改和新增
    2.4 关于Docker+Play技术栈的并发优化
  欠款：
    2.5 设计模式的实际应用

学习与培训纪要

博客地址: Pharbers用于单点登录的权限架构
博客地址: 应用篇 = Docker下的Redis

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代

1. 类Shiro的登录培训

Your Inbox => 幸存区

1. 十万并发下的兔子问题(斐波那契数列)怎么做
2. 阴险的约瑟夫
3. APM计算代码重写
4. Docker 的 Redis 安全问题博客

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	配合森哥调试新接口, 没有问题则发布到docker	90%	如果要加密的文本长度为m, 同时按照512bit加密, 则m的长度应该 0 < m < (512/8 - 11)		下一步	
Tuesday周二	和森哥对接登录接口	100%			归档	
Wednesday周三	和森哥对接用户注册接口	100%	修复无数小bug	登录和注册, 看着简单, 实则细节很多, 并且为了之后的灵活修改(跟得上产品步伐), 所以也耽误了一点时间.	归档	
Thursday周四	1. 为了更加安全可靠的数据查询, 之前的bind_sales表拆分, 加入time_type 2. 稍微的改一下scala版本的APM计算	1. 100% 2. 100%	(之前设计的不好, 导致现在做了体力活) APM计算改完了, 但是数据不对		下一步	
Friday周五	1. APM计算对数 2. 配合安琪, 森哥修改bug	1. 100% 2. 100%	数对不上, 在安琪的指导下, 双方验证, 我们有整型除法的精度问题, 同时R也有些问题, 耽误很久	做事一定要细心, 要稳重, 真的.	归档	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

做事一定要细心, 细心, 细心, 更要稳重!!!
做事一定要细心, 细心, 细心, 更要稳重!!!
做事一定要细心, 细心, 细心, 更要稳重!!!

学习与培训纪要

1. 关于Docker下的MongoDB备份和还原:
   备份: 第一步, 连接到docker宿主; 第二步, 进入mongodb的container, “docker exec -it fc0dc2fa21d3 /bin/bash”; 第三步, 执行备份, 为了方便copy移植, 备份到外部映射文件中, “mongodump -h 127.0.0.1 --port 27017 -d pharbers-client -o /data/dump”
   还原: 第一步, 连接到docker宿主; 第二步, 进入mongodb的container; 第三步, 执行还原, “mongorestore -h 127.0.0.1 --port 27017 -d pharbers-client -o /data/dump/pharbers-client”

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox => 新生代

1. 类Shiro的登录培训

Your Inbox => 幸存区

1. 十万并发下的兔子问题(斐波那契数列)怎么做
2. 阴险的约瑟夫
3. APM计算代码重写
4. Docker 的 Redis 安全问题博客

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	重写杨总的RSA加密算法及其加密相关的数据库设计,	90%	如果要加密的文本长度为m, 同时按照512bit加密, 则m的长度应该 0 < m < (512/8 - 11)		下一步	
Tuesday周二	1. 继续加密问题, 我和老铁互相解密. 2. 密文的中文字符和长明文循环加密问题.	95%	1.还是加密问题, 我和老铁生成的公钥和私钥都是PKCS1PADDING规范的, 但是老铁加密的密文是PKCS1规范, 而我的Java端, 只有PKCS8规范. 2. 同时, 在我和鹏哥进行中文测试和长明文需要循环加密, 测试的时候出现问题, 前端JS没有Bytes的概念.	任何计算机问题, 都可以通过加一层中间件来解决. 所以鹏哥提出一个想法, 就是将明文先进行一次URL编码, 然后在进行加密. 可行性明天测试.	归档	
Wednesday周三	1. 加密问题结束 2. DES加密 3. 角色数据库设计	1. 100% 2. 100% 3. 100%			归档	
Thursday周四	可单点登录的权限系统实现	100%			归档	
Friday周五	因为新的token加密内容, 以前节点需要重新编译部署	60%	**问题倒是没有, 但这完全是当初可以避免的无用功, 体力活.**	本来是不需要这一步的, 以后记住, 没有应急方案, 没有测试版本, 你怎么对代码, 代码就怎么对你, 所以, 好代码, 不将就.	下一步	
Saturday周六	继续部署和测试	100%			归档	
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

学习与培训纪要

1. 最近被加密折磨的很痛苦, 尤其是周二晚上和鹏哥对接中文以及长明文分段加密的时候, 所以在此记录了一下关于所有常见编码的起源及编码方式https://blog.csdn.net/zxh2075/article/details/53064160
   ASCII => GB2312 => GBK => GB18030 => ISO-8859-1(UNICODE) => UTF-8(UNICODE)

吃饱了撑着没事儿闪现的Idea	1. 以后记住, 没有应急方案, 没有测试版本, 你怎么对代码, 代码就怎么对你, 所以, 好代码, 不将就.
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务

帮助大家应用「GTD」进行时间管理

请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox
1. 十万并发下的兔子问题(斐波那契数列)怎么做
2. 阴险的约瑟夫
3. Docker 的 Redis 安全问题
4. Shiro的思想摘要

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一						
Tuesday周二	将APM所有节点提取出来, 以解决APM并发的死锁问题	100%	完全的体力活, 搞定了 https://www.getpostman.com/collections/8bbc81aceeaef7148db1		彻底搞定了	
Wednesday周三	调研Shiro问题, 找寻适合PH的方案	1. 30%	shiro在功能上和安全上, 都是上个年代相当成熟的产品, 想要`copy`出来还是有难度的.		下一步	
Thursday周四	1. 调研Shiro问题, 找寻适合PH的方案 2. APM调R低效的问题+	1. 70% 2. 100%	2. R低效的问题已经找到, 主要在于R代码对于插入report时, 因为无法获取新的id, 需要全表遍历，取最后一条数据的id(这个解决办法真的是...), 所以初步决定, 由安琪提供算法, 我们Scala重新实现, 暂定完成, 触发开关, 产生新任务. => APM计算代码重写	APM计算代码重写	这事儿已进入下一个环节	
Friday周五	1. 安琪新的数据, 记得换上. 2. 查询优化(如paper倒序取20, 其余为"more")	1. 100%			彻底搞定了	
Saturday周六						
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价

Shiro的功能很全, 但我们没必要全搬全抄, 毕竟我们有Redis和OAuth

学习与培训纪要

本周没有写笔记, 先欠着两篇博客
1. Docker 的 Redis 安全问题
2. Shiro的思想摘要

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

OKR Weekly Report Lev.1-5

计划与任务
帮助大家应用「GTD」进行时间管理
请大家花15分钟之内，可以将任何「已知/未知」的「事务/任务」、「想法」、「念头」，写进下面这个「Inbox盒子」

Your Inbox
如果在你写的时候，发现这件小事儿基本花「2分钟」就可以完成，那么写完就马上去做，做完就删掉这个事儿

然后将这些个事儿，组织安排进下面的OKR Weekly小表格中，并在需要的时候进行描述

	描述下「你要做的事儿」	Situation **推进的情况**	**如果遇到**「冲突、挑战与问题Challenge Complication Question」	**试图寻求或者思考的**「解决方案Solution & Answer」	**GTD标注**	
Monday周一	「项目/产品」				删除	彻底搞定了
Tuesday周二					归档	需要归纳起来
Wednesday周三					下一步	这事儿已进入下一个环节
Thursday周四					将来可能	停下来，放在以后
Friday周五					等待	等等，我需要其他支持才能继续
Saturday周六					项目	成为一个更大事儿的某个后续组件，写出
Sunday周日						

随着时间的变化，大家可以对自己预期规划的事儿进行跟踪记录与调整检查

如果有新的事情插入，先对事情的轻重缓急做一个简单的判断，根据已有的事情进行穿插安排，或者直接扔进Inbox先

最后，在一周结束的时候进行相关事务/任务的总结，对每一件事情进行「GTD」标注，并准备下一周的安排

思考/自我评价
如果你有任何自我思考、困惑或者反思回顾，那就麻烦你想写什么就写点

学习与培训纪要
如果你想写点有关学习与培训的事儿，那你提炼一下知识内容

吃饱了撑着没事儿闪现的Idea	
狠狠提意见	

表格 1

6 Weeks	**OKR Planning Meeting**																	**OKR Review Meeting**				
2 Weeks				**Scrum Planning Meeting**											**Scrum Review Meeting**							
1 Week							**Weekly Planning Meeting**						**Weekly Review Meeting**									
Daily										**Standing Up Meeting**												
	T Group	P Group	M Group	T Group	P Group	M Group	T Group	P Group	M Group	T Group	P Group	M Group	T Group	P Group	M Group	T Group	P Group	M Group	T Group	P Group	M Group	**碰头会**	**头脑风暴会**	**展示会**
目的	每6周为一个节点，让大家清晰地了解「阶段目标」，并了解为达成目标所需要的「关键过程或结果指标」。以此朝着目标推进。			敏捷项目开发/管理的目的是：适应需求的变化，缩短适应市场的周期。			团队成长	团队成长	周会，是为了计划、追踪业务目标达成的路径和解决相关的问题				团队成长	团队成长	周会，是为了追踪业务目标达成的路径和解决相关的问题	敏捷项目开发/管理的目的是：适应需求的变化，缩短适应市场的周期。			每6周为一个节点，让大家清晰地了解「阶段目标」，并了解为达成目标所需要的「关键过程或结果指标」。以此朝着目标推进。			快速沟通不同Group之间的协调性	解决创造性问题	向团队或外部人员进行演示和阐述
负责人	Tech lead	Supervisor	Supervisor	Tech lead	Supervisor		Mentor	Mentor	Group lead	Group lead	Group lead	Group lead	Mentor	Mentor	Group lead	Tech lead	Supervisor		Tech lead	Supervisor	Supervisor			
维护者	OKR指定轮值负责人			Group lead	PM		Mentor	Mentor	Group lead	Group lead	Group lead	Group lead	Mentor	Mentor	Group lead	Group lead	PM		OKR指定轮值负责人					
会议流程	解读「公司目标」			确认Sprint Backlog	持续规划和编制Product Backlog					昨天做了什么?						回顾Sprint Backlog执行中的问题	持续规划和编制Product Backlog		回顾「所有目标」完成情况					
	阐释「部门目标」			「What」拆解Task; 「Who」分配Task	提供「交付物」					今天打算做什么?						「How」回顾Task完成情况;「Why」分析遇到的问题	持续改善并提供「交付物」		分析反思遇到的「所有问题」					
	探讨「挑战」和「关键过程及结果」			「When」明确Deadline						有什么问题?						向团队成外部成员进行展示								
会议议程	有	有	有	有	有				有						有	有	有		有	有	有			
会议纪要	有	有	有	有	有				有						有	有	有		有	有	有			
时间	周五 15.30			周二 16.00	周一 16.00		周五	周五	周一 15.00	12.45	11.00	10.00	周五	周五	周一 15.00	周二 16.00	周一 16.00		周五 15.30					
频次	Every 6 Weeks			Every 2 Weeks	Every 2 Weeks		Mentor自行安排	Mentor自行安排	Every Week	Daily	Daily	Daily	Mentor自行安排	Mentor自行安排	Every Week	Every 2 Weeks	Every 2 Weeks		Every 6 Weeks					
控制	3-4 hours			1 hour	1 hour		1 hour	1 hour	1 hour	15mins	15mins	30mins	1 hour	1 hour	1 hour	1.5 hour	1.5 hour		3-4 hours					