

Name<齐钟昱>

Group<Max_Dev>

Project<Max>

Pharbers Monthly OKR List No.<1>

Date 2017 09 :

Update By <09-18>	Update By <09-24>	Update By <09-29>
Update By <09-19>	Update By <09-25>	Update By <09-30>
Update By <09-20>	Update By <09-26>	
Update By <09-21>	Update By <09-27>	
Update By <09-22>	Update By <09-28>	

Date 2017 10 :

Update By <10-09>	Update By <10-16>	Update By <10-25>
Update By <10-10>	Update By <10-17>	Update By <10-26>
Update By <10-11>	Update By <10-18>	Update By <10-30>

Update By <10-12>	Update By <10-19>	Update By <10-31>
Update By <10-13>	Update By <10-20>	
Update By <10-14>	Update By <10-23>	
Update By <10-15>	Update By <10-24>	

Date 2017 11 :

Update By <11-01>	Update By <11-12>	Update By <11-20>
Update By <11-02>	Update By <11-13>	Update By <11-21>
Update By <11-03>	Update By <11-14>	Update By <11-22>
Update By <11-06>	Update By <11-15>	Update By <11-23>
Update By <11-07>	Update By <11-16>	Update By <11-24>
Update By <11-08>	Update By <11-17>	Update By <11-27>
Update By <11-09>	Update By <11-18>	Update By <11-28>
Update By <11-10>	Update By <11-19>	Update By <11-29>

Date 2017 12 :

Update By <12-04>	Update By <12-13>	Update By <12-24>
Update By <12-05>	Update By <12-14>	Update By <12-25>
Update By <12-06>	Update By <12-15>	Update By <12-26>
Update By <12-07>	Update By <12-18>	Update By <12-27>
Update By <12-08>	Update By <12-19>	Update By <12-28>
Update By <12-09>	Update By <12-20>	
Update By <12-11>	Update By <12-21>	
Update By <12-12>	Update By <12-22>	

Date 2018 1 :

Update By <1-02>	Update By <1-11>	Update By <1-22>
Update By <1-03>	Update By <1-12>	Update By <1-23>
Update By <1-04>	Update By <1-13>	Update By <1-24>
Update By <1-05>	Update By <1-15>	Update By <1-25>
Update By <1-06>	Update By <1-16>	Update By <1-26>
Update By <1-08>	Update By <1-17>	Update By <1-29>
Update By <1-09>	Update By <1-18>	Update By <1-30>
Update By <1-10>	Update By <1-19>	

Monthly Personal OKR

Objective 目标1：提升编程技能，不欠技术债（**完成**）

KR1 继续看书《Scala编程》，每周5章（**不够具体**）

KR2 看书Gating压力测试工具，学会看测试报告（**没有时限**）

KR3 系统学习MongoDB，并搭建分片集群（**没有时限**）

KR4 撰写读书笔记和思想感悟（**完成**）

Objective 目标2：替换Max中生成panel的算法，配合MAX上线（**其上有个更大的**

Objective=>完成Max的商业化上线）**整体而言，不想是个O，更像是个KR**

KR1：提高Scala对处理非标准Excel的健壮性（**没有时限**）

KR2：在算法和数据源正确的情况下，保证生成的panel文件完全正确并且可用（**没有时限**）

KR3：完成与Max的对接工作,以jar方式引入并根据设定的参数可正确生成panel文件（**没有时限**）

KR4：根据算法实现panel文件的时间选择功能（**没有时限**）

KR5：重构panel文件生成代码（**没有时限**）

KR6：修改panel算法，支持单月份单市场输出（**没有时限**）

KR7：提升panel算法的写入速度（**没有时限**）

Objective 目标3：完成Max的对接，完全替代Python算法**整体而言，不想是个O，更像是个KR**

KR1：学习Max的calc的代码和流程，并在本机运行成功（**没有时限**）

KR2：完成上个sprint中我和张弛工作成果的对接（**没有时限**）

KR3：使用堆转储技术对Panel生成算法进行内存优化（**没有时限**）

KR4：提取calc中的内存管理 (没有时限)

Objective 目标4：掌握Actor的设计模式 (不具体，无法衡量，没有时限)

KR1：第二周学习并掌握Extra设计模式

KR2：第三周学习并掌握Cameo设计模式

Objective 目标5：为复杂但稳定的计算流程加上漂亮的衣服整体而言，不想是个O，更像是个KRs

KR1：一周内完成sprint03中client和calc对接，并通过测试

KR2：Max内测版本上线 (不具体，没有时限)

KR3：Max新版本的页面拼接 (不具体，没有时限)

KR4：清洗Max老架构代码，降低以后人员的学习成本 (不具体，无法衡量，没有时限)

Daily Brief

Date: 2017 09 第_3_周 0918-0924

日期	目标 Objective	关键结果KR	工作项目	时间 颗粒 预期	完成用 时	完成情 况	自我回顾 及评价
Mon	替换Max中生成panel的算法，配合MAX上线	1.完成panel生成算法的实现	1.对于非标准Excel数据可以正常读取 2.和python算法生成一致的文件	1 day	2 day	done	本次实现主要出现以下问题：1.对于单元格中None，“”，#A/N的处理没写好，导致数据读入错误（易被发现，已解决）2.单元格中存的数字分为字符串和整型，使用默认的读取方式就会导致以0开头的id无法匹配（不易发现，已解决）3.对于生成panel的源数据文件更新不及时，导致每次生成的文件数目对不上（团队的内耗是一定的，但是要将这种没有意义的消耗降到最低，而直接的沟通就是最好用的方式）
Tue		1.测试算法正确性	分别测试辉瑞201611到201705的数据源并与正确数据比对要求完全一致	0.5day	0.5day	done	1.生成结果经常对不上，经过和彭总的沟通了解是因为我们双方数据没有及时更新的问题（哎，怎么说呢，感觉这种问题完全可以避免，但经验不足使我在Pharbers的第一次沟通形式的开发中浪费了很多时间，也走了许多弯路，吸取的经验就是尽量确定自己功能正确的同时也不可完全肯定别人提供文件的正确性）
Wed		1.查找算法不匹配原因	1.将算法以jar形式引入Max_calc中 2.学习Max代码并在合适位置替换原python代码 3.查找panel文件不同原因	1day	1day	done	1.panel不一致原因： 小陆提供的线下版本删除了匹配不到ims pack id的药品，所以导致使用线上文件生成的panel多出35条,但是后期小陆会重新加上这些药品。
Thu		实现panel文件的时	1.根据昨天比对结果小陆生成了最新的线	0.5day 0.5da	0.5day 0.5day	done	今天正式接触了calc项目，理清了项目类的加载关系，并且在鹏哥的帮助下完成了对我这两周实现接口的简单调用。

		间选择功能	下版panel文件，现与Scala生成的完全一致 2.panel有一个补充的时间选择算法今天完成了	y			但是在给鹏哥演示的时候发现了算法不完整，经过他的协调和讲解我理清了项目流程，并完善了整个算法。 不得不说这两天的比对过程中，我对Excel的使用真的进步很大，以前还真是小看了这一类的工作软件。 最大的收获当然就是看着杨总和鹏哥在重构挠头代码了，自己提心吊胆的同时想着偷偷的优化一下自己的panel，代码质量才是检查一个程序员好坏的最直接指标，对公司负责，也对自己负责。
Fri		1.接口对接 2.压力测试	1.重构panel代码 2.学习calc项目，完成对接 3.对panel文件生成进行压力测试	4h 3h 4h	3h 3h 0h	done	不知道在上班时有时间学习是不是好事，但是今天的任务还是做的很快的。 今天最大的收货是对于OKR的更加了解，书写更加专业了。OKR的内容对于我来说还是过于抽象，不是能很好的理解KR和Task。但是有个非常简单和不变的原则就是，“先明确有具体导向的O，在划分有时间粒度和具体数据的KR，来使Scrum可以度量员工效率”
Sat							
Sun		1.对panel文件生成队列进行压力测试	同时发送50个生成201705年的panel文件的请求到队列中，统计执行结果和结果正确性	4h	1h	done	周五的下班的时候运行发生错误，所以测试时间推迟到今天，本以为对于项目不熟悉，会调试很久，还好只是一点小问题，几次调试就解决了。本次测试结50个全部通过，结果也全部正确，每次最多可运行四个，50个请求总运行时间5.5小时，平均每个请求耗时6min，总体来说还是可以的。

Date: 2017 09 第_4_周 0925-0931

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	合并max分支，完成client中计算页面以配合MAX上线	完成计算页面	1.导入client工程并可以正常运行 2.根据线上max完成本地计算页面	2h 2.5day	1day	完成工程导入	本来以为可以非常顺利的和大家同步client项目，并顺利的运行，但是项目一直无法启动，最终在杨总和鹏哥的帮助下成功运行，无法运行的原因是MongoDB相关的jar包版本在我的电脑上不支持，中间走了许多弯路，因为一直没怀疑是这个问题，因为除了我以外的其他人都是可以正确启动，最终还是在杨总的大胆假设小心论证下拿下了这场战役，受教了受教了。
Tue		完成计算页面	1.学习layui 2.cpa，gycx导入 3.样本检查数据查看页面的三个图表	4h 2h 4h	3h 1.5h 5.5h	图表完成四个	图表是大数据展示给用户最为友好的方式了，我现在还不用考虑对对图标样式的选择等等，但是光是渲染图表所需要的数据结构就非常麻烦，本来不想抄原来工程代码的，要是明天还不能解决，就只能抄代码了。:-)
Wed		完成计算页面	1.样本检查数据查看页面的第四个图表 2.样本检查页面的表格 3.结果查看页的三个柱状图	1h 2h 4h	1h 4h 2h	done 发现一个bug，eachats加载的图表如果是首个加载的tab正常显示，不是第一个的话就无法加载	在表格处理上浪费的极多的时间，首先layui为了减少开发代码，所以对于table的数据渲染推荐采用的是直接使用uri加载，因为现在是开发过程，所以自己伪造假的数据，存到js变量中，而table的格式我是在html中定义的，这种奇葩的混搭方式使得数据无法渲染到表格中，各种方式尝试后，只能将table的格式定义在js中解决这个问题，但是table就无法进行适应屏幕宽度进行伸缩。目前官网也没有合适办法(官网对于table的例子也是挺丑的)。

Thu		优化计算页面	1.和郭人豪沟通计算流程 2.优化计算流程页面 3.学习前端规范，重构前端代码	0.5h 3h 5h	0.5h 3h 4h	done	js学起来简单，但是完全按照鹏哥给的前端构建规范来编写还是有一定挑战的。 其中js的定义域最是烦人，js的函数式编程也很不好掌握，最容易产生错误的地方就是js语法了，太灵活了，而且不好报错，他的加载方式也会导致页面产生一些无法理解的错误。 以下几点不理解的地方， 1.命名方式为什么使用-连接而不是_，下划线才是命名标准啊，也可以被IDE识别（ 纠正，只有id以-命名 ）。 2.js异步加载很方便，为什么其他工程都没使用， 附1：动态加载时序图 3.许多来源的库的确是采用的IIFE(Immediately-Invoked Function Expression译为“立即调用函数表达式”) 方式避免全局变量污染，但是加载方式和使用方式没有原生态的方便，是方式不对还是根据实际取舍。 以上三个问题留待以后更加专业再回来解答
Fri	Max新架构测试	1.理解并使用Max新架构流程和代码 2.黑盒测试Max 3.白盒测试Max 4.修改Max代码	1.分别使用黑盒和白盒测试Max系统的注册流程，激活流程，登录流程和忘记密码流程	1day	1""day	done	体验了一次测试人员的“辛劳”和“艰辛”，简直是飞一般的感觉，但是我知道我的“幸福”是建立在别人的“痛苦”之上的，所以，为了自己少点痛苦，每一行代码都要仔细仔细再仔细，要注重代码质量，以提高代码的维护效率，不然每次上线都是噩梦啊。 附，Scrum_Max第一轮黑盒白盒测试结果
Sat		主要针对业务流程和用户体测试	1.以用户的角度，身临其境的对Max进行注册和使用	1day	1day	done	
Sun							

Date: 2017 10 第_2_周 10/09-10/15

日期	目标 Objective	关键结果KR	工作项目	时间 颗粒 预期	完成用时	完成情况	自我回顾 及评价
Mon	一周内完成Max的对接，完全替代Python算法	1.学习Max的calc的代码和流程，并在本机运行成功 2.完成上个sprint中我和张弛工作成果的对接	1.在本机完整的跑一次calc流程 2.和张弛对接成功	2h 4h	未完成 未完成	本机运行失败，原因正在查找 接口对接失败，原因正在查找	就像那个传说中的神奇的定律一样，任何事情都比想象中的要困难，本来认为很简单的事情总能出现一些不简单的问题，还好杨总经验老道，知道我们实力，也让我没有夸下海口，再一次的认清了自己的实力。 刚接触akka的设计模式，现在一头雾水，有一些看不见摸不到的思维在脑海里，但是还无法用语言来描述出来。
Tue			1.在本机完整的跑一次calc流程 2.和张弛对接成功	1h 3h	1h 未完成	本机运行计算成功 接口对接失败，正在查找原因	总结昨天接口对接失败原因： 1.我生成的panel文件为2003版本的Excel文件，而计算需要的是2007版的(已解决，当初实现算法的时候只顾着结果正确性，没和这面沟通，还好修改简单，但是浪费了大家的时间) 2.新的生成算法在单元测试时候没有问题，且可以生成计算代码可用的panel文件，但是当以jar包形式引用到程序中时，调用新的poi的api会报错，没有找到方法(正在查找原因)
Wed			完成和张弛的对接	1h	未完成	生成的panel文件calc无法读取	今天解决了昨天的第二个问题，但是昨天好使的panel文件今天不好使了，原因不明，正在杨总的指导和协助下寻找解决办法

Thu			1.将中间文件panel由Excel改为csv 2.完成和张弛的对接	2h 6h	2h 未完成	目前整套流程可以跑通，但无法并行	收获: 1.上午，跟张弛学习了工程中用到的Cameo设计模式，可以说不是很难，但是非常麻烦和复杂，其中也能看出杨总在最初设计时对系统的健壮性以及扩展性方面的考虑 2.中午，杨总讲解了calc项目对于内存的抽象和优化的相关代码，这段代码是我刚来法伯就接触的code，今天杨总讲完我才有一种登堂入室的感觉，并且才明白什么是真正的程序，什么又是真正的程序员，这就是为什么 不轻易 使用插件、外部库和图形界面的原因，这种掌握每个细节的感觉太爽了 3.傍晚和鹏哥讨论js的一些语法，句式，和一些比较模糊的问题，因为和本周o关联不大，不再赘述
Fri			1 重新学习GC知识 2 学习jvm检测工具VisualVM 3 查找程序中造成内存泄露的原因	2h 1h 6h	2h 0.5h 7h	done	昨晚进行全流程单行的压力测试，运行50个，在运行第四个的时候报错《内存超出》，学习Java的GC相关知识和JVM检测工具VisualVM的使用，最终在杨总的帮助下查找到内存超出的主要原因是我写的panel生成算法没有考虑内存问题，还有一些代码的细节浪费了一些内存，但是主要原因是加载的数据量太大，如果生成pane的同时进行calc，那么在某次峰值的时候就会内存超出，所以关于这部分代码要进行一些特殊处理----堆内存转储 附4：《开发过程中内存优化的几点建议》
Sat		使用堆转储技术对Panel生成算法进行内存优化	1.学习数据结构 堆 的相关知识 2.尝试使用Scala实现堆转储	4h 4h	4h 未执行		
Sun			尝试使用Scala实现堆转储	4h	5h		如果已有的数据结构可以实现我们需要的效果，那就不需要重新定义新的数据结构了啊，所以就不能用List为基础定义一个新的完全二叉树，而且既然是内存管理，就不应该将结构定义在内存中，而是应该在文件中并且可以和文件进行简单的通信交互。

Date: 2017 10 第_3_周 10/16-10/22

日期	目标 Objective	关键结果KR	工作项目	时间 颗粒 预期	完成用 时	完成情 况	自我回顾 及评价
Mon	替换Max中生成panel的算法，配合MAX上线	1.修改panel算法，支持单月份单市场输出 2.提取calc中的内存管理	1.修改panel算法，支持单月份单市场输出 2.将calc中的alJob包提取出来形成一个单独的工程，方便重用	4h 5h	6h 未完成	done 还有一些多余代码没有删除	根据领导需求，panel算法改为传入一年中全部市场的所有数据，而生成的为可多选的月份中的全部的市场生成panel,完成还算顺利，主要是要和鹏哥核对接口。没有按时完成任务的主要原因就是在类型转换的地方，虽然每次都会感觉到对Scala类型系统掌握的进步，但是每次遇到问题的时候都会消耗一些时间，真的只能说，有JsValue的地方实在是太麻烦了。
Tue			1.提取calc中的内存管理 2.采用新的nio方式对panel算法进行内存优化	2h 12h	5h 未完成	done	本来以为看懂了代码，对于memory的代码移植会非常简单，但是事与愿违。这次重构的Memory模块包结构和之前的不一样，所以调整了很多代码，还有一个好多地方都用到的特质类，我处理的不好，净想些花里胡哨的东西，对于重构没有经验啊，浪费时间。然后测试时候，没有调整代码，关了几个软件清理了一下系统环境，第三次测试终于通过了。
Wed		采用新的nio方式对panel算法进行内存优化	1.学习延迟加载思想和Steram 2.采用延迟加载对算法进行内存优化	3h 5h	3h 未完成	新的panel实现完成80%	首次接触延迟加载和延迟计算，又是一次新的体验和成长。虽然Scala已经将api封装的很完美了，但是在思想上和函数式编程还是有些细微差别的。延迟加载，之所以这么叫，是因为它在初始化构建的时候就只是构建了第一个data，而之后的数据使用一个function表示，这个在使用调试的时候可以watch一下，Stream[T]=T+function,很新颖的想法，而我还没有完全的脱离Java编程思想，因此又受到了杨总的"鼓励"，嗯，再接再厉。 关于今天遇到的一个小问题，我把一个延迟计算的方法的函数值赋值给了一个val导致编译出错，解决办法在下面链接 https://stackoverflow.com/questions/22108579/cal-ling-functions-before-they-are-defined-forward-reference-extends-over-defini 也就是改为def或者是lazy val

Thu			1.学习 MappedByteBuffer文件写入方式 2.利用缓存文件优化算法	4h 5h	5h 7h	95%	1.附5：copy and swap思想 2.附6：MappedByteBuffer双缓存技术 3.对于Java到Scala思想的转变，Java多使用接口驱动，而Scala没有接口，而是引入了特质的概念，既然名字不同了，用法自然不同，可以利用细粒度特质的不同组合更大程度的重用代码并且更灵活的对功能进行管理 4.看了张弛分享的一篇文章，深度分析算法的产物AlphaGo Zero左右互搏了三天就打败了AlphaGo三个月学习完人类三千年积累下来的智慧结晶，虽然棋局的中段和人类下棋思想有所偏差但是后面思路还是殊途同归的。人类用自己创造的工具证明了自己思想上的正确性，同时也找到了一些曾经未被涉及到的思维漏洞，这本是值得高兴的事情，AI是用来服务人类的，我们不能限制她，可是我也感觉到了AI的可怕，思细极恐，希望霍金的预言永远不要到来。
Fri			1.对panel文件进行大小限制，保证每个文件在300KB一下 2.在源数据表中添加新增的数据，并更新panel算法到v2	2h 5h	8h 未完成	生成的panel数据对不上，原来是数据部的算法更新了	为啥打算2h收尾的工作做了8h呢，哈哈，因为之前有一条数据异常，我debug的时候，使用条件断点(类似assert，但是效率慢很多，很多很多)，然后我忘记自己有这个断点了，idea中也没有显示小红点，之后就悲剧了，文件的写入速度慢很多（断点在这部分代码），我以为算法有问题呢，后来在杨总的指导下，设计的思路是对于单文件使用二分查找优化插入速度，然后多文件merge并进行自定义处理，这样的好处是更符合FP思想，代码分工明确，每个function只做一件事，并且杨总眼光更远，这样的设计更利于向分布式靠拢。可是就在我该代码的过程中，断点意外的去掉了，效率目前也不是问题了，问题解决了，虽然走了弯路但是也因祸得福了，但是在此给各位提个醒，编程还是要细心一些，不然真的很耽误事啊。
Sat							
Sun							

Date: 2017 10 第_4_周 10/23-10/29

日期	目标 Objective	关键结果KR	工作项目	时间 颗粒 预期	完成用 时	完成情 况	自我回顾 及评价
Mon	完成算法2.0版本	1.根据提供的算法进行数据补充 2.和张弛完成对接，并可以承受规定的压力测试	1.与张弛进行完成对接，并可以100%通过50个压力测试 2.和运营沟通，了解最新的数据情况，并修改panel生成策略	2h 2d 6h	6h ToDo	对接完成，但是压测不稳定 已经沟通完成	本来任务就已经拖延了很久了，还未对自己的代码进行压测，导致出现文件打开数量过多的系统级错误，还好之前遇到过，已经解决。 但是目前还是无法通过压测，具体原因还在杨总的帮助下排查，正在尽最大的努力解决该问题。 不知不觉的又拖慢了团队的进度，事情发生了自责也没用，只能在本次错误中总结错误，避免同样的错误再次发生。以后必定全力以赴，将功折过。
Tue			1.重构代码，在杨总发现前擦好上周的尾巴 2.完成v2版本的补充cpa未到医院数据	4h 4h	6h 2h	基本完成，但是提供的医院数据乱码，明天重新要一份	在昨天的沟通中发现需求对于panel的生成流程有了新的要求，而这次的改进正好去掉了一个之前比较麻烦的地方，正好这段时间对于scala的编程思想有了新的体会，scala是组件式的开发，并且对于一套任务流程，讲究的是一步一个脚印，所以决定抽时间对算法进行重构，方法更加细粒度并且流程可读性更高，方便之后的人员维护优化。
Wed			1.和需求测试算法的正确性 2.检测最新算法的资源占用情况 3.和calc对接，完成压力测试	4h 3h 1h	4h 3h 1h	done	延期一周，正式完成杨总交代的全部任务，回顾过去两周，总结一下几点： 1.首先，基础很重要，我在最初版本的panel算法中，只考虑了算法的计算效率，而忽视了内存方面，而且因为没有做过内存管理，在之后的学习中也相对较慢，这导致了我们的全部多了一周的工作量，罪过罪过。 2.延迟加载，mappenByteBuffer，这种技术平时就应该掌握，而不是临时抱佛脚 3.因为debug操作不当，导致的杨总以为最初想法有误，重新制定内存管理的一些细节，导致浪费整整一天（其实当时已经好使了） 4.排序，查找，树，堆，内存，寻址，这些最最最基本的！别以为知道就行，必须透透滴 5.有啥事直接沟通，才是解决问题最根本最直接的途径，需求给的文件乱码，我竟然想着改程序，就不会让人家直接生成一份吗，笨蛋

Thu	打杂	1.辅助杨梅 sprint2 2.max计算的多市场问题 3.准备培训	1.配置最新的client代码环境 2.完成client的任务 3.和需求统计可计算的所有市场，生成后核对各市场结果，并完成和张驰的对接 4.准备培训前两周学习的内存管理	0.5h 3h 3h 1.5h	3h 3h 4h 未培训	培训顺延至下周	<p>在需求的大力配合下，再次修改了算法，还好在之前杨总传授的编程思想下，本次修改流程速度很快，这是在自己编码能力上和思想上一次很大的提升，程序员就应该做好代码的模块分离和可重用，一个方法一件事，只有写出可维护的好代码，才能适应不断改变的新需求，想象着我也可以说出那句豪言壮语“就改这个啊，我两行就搞定”，好的代码改起来看起来真的是很爽。</p> <p>目前可以生成INF和SPE两个市场的panel，本来应该很快就可以测试完的，可是要下班了，干啥都着急，我自己测试的时候不全面，没有考虑分隔符问题，导致浪费了自己和张驰半个小时的时间。程序员偷懒可以，但是蠢可真是要不得。</p>
Fri							
Sat							
Sun							

Date: 2017 10 第_5_周 10/30-11/5

日期	目标 Objective	关键结果KR	工作项目	时间 颗粒 预期	完成用 时	完成情 况	自我回顾 及评价
Mon	完成 sprint2 回顾检 查	1.在郭人豪的帮助下梳理两种注册流程，并且根据各种情况对数据库内容进行取舍，以兼容所以可能的bug	1.用户申请流程 2.bd预约流程 3.bd入库并发送验证码 4.bd帮助填写信息直接注册 5.第三步第四步的冲突处理 6.闭环测试	1h 1h 1h 1h 3h 1.5h	3h 1h 1h 1h 3h 2h	第一个任务超时是因为还有对源代码的学习成本 闭环测试时发现新bug，所以耗时增加	刚接触这次sprint的client部分，逻辑相当复杂。而且不是自己写的代码真的很难维护，
Tue		1.对已有全部功能进行测试 2.sprint02总结会议和sprint03任务分配	1.解决token过期功能bug 2.解决主页中显示的个人信 息bug 3.和郭人豪一起测试全部流程 3.和钱鹏学习如何优雅的处理json和pr 4.sprint02总结会议和sprint03任务分配	1h 0.5h 3h 2h 2h	0.5h 0.5h 3h 1h 2h	done	比较顺利，不想写啥了。 在今天的测试中， sprint02中未通过的功能 见附录。
Wed	为复杂但稳定的计算流程加上漂亮的衣服	一周内完成sprint03中client和calc对接，并	1.测试最新分支中鹏哥原本完成的计算流程并学习 2.完成后台的生成panel流程的接口	3h 3h 4h 3h	1.2h 2h 3h 4h	80%	现在后端接口已经完全符合业务流程，也就是要发送三次http请求，分别筛选年份，生成panel，和真正计算。页面需要上传源数据来进行计算而不是panel，目前唯一的小bug就是我和钱鹏数据库不一样，所以在计算的时候，传入的公司为空，计算失败，明天就可以跑通全部流程，进行细节优化。之所以完成的比预期要快，主要原因就是因为钱

		通过测试	3.完成前端的文件上传功能 4.统一文件位置，并调通前后端				鹏代码写的规范，而且易于维护，这就是我之前多次提到的，要写好代码。 最后整合的时候，浪费很多时间，主要是对js的api不熟练，还有一些其他简单的小事情，耽误了很多时间。
Thu			1.测试接口的稳定性 2.和钱鹏学习和探讨环信在本系统中的模式 3.关于环信聊天室功能的学习和思考	3h 2h 3h	4h 1h 2h	done	今天算是来pharbers半年来第一次和同事进行这种正式的技术讨论，发现我以前考虑的都是对当下情况的最优解，而杨总会把复杂的问题抽象为简单的情景，在以未来进行分布式为可能而编写代码。钱鹏则会为所有的业务模块想好未来可能的业务逻辑，事先就留好代码的提升空间，既方便了未来业务的扩展还能保持接口不冗余。值得学习。
Fri		利用昨天钱鹏设计好的环信聊天室功能，拼接页面，争取周一做出一个版本。	1.对计算页面的环信回调部分进行代码重构。 2.添加一个tab为源数据上传 3.更改后台返回的数据格式和消息类型，拼接逻辑 4.链接到鹏哥上周的模块上	30min 1h 2h 1h	5min 3h 4h 3h	理论上已经拼完了，但是panel不生成，明天解决（周六已解决）	页面要做好，我还有很长的一段路要走，而且每次我预估时间都是按照最优情况，但是每次执行任务的时候，都会遇到一些未曾考虑的问题，如环信同时上传多个文件时，完成的回调函数不是返回一个有多个值的数组，而是回调响应多次，每次返回的数组中只有一条数据。 还有代码编写完成之后，测试的panel生成不通过，后来找到原因是因为在没开启全部节点的时候，代码中关于查找互斥节点的位置，相应的要修改一些代码，
Sat							
Sun							

Date: 2017 11 第_1_周 11/6-11/12

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	Max 内侧版本上线	完善并测试项目稳定性	1.加入 panel 进度条实时回调 2.加入聊天室 创建逻辑 3.测试目前项目稳定性	1h 2h 4h	1.5h 2h 5h	测试中	主要就是对系统稳定性进行测试，总的来说没遇到什么大问题，但是还是有些麻烦事的，比如为了解决环信同一账号登录出现的挤掉问题，使用了聊天室功能，而聊天室的状态同步却很是不好控制，同时还有一些同账号的操作限制等，都需要重新考虑一下。
Tue		根据郭人豪给的产品原型拼接前端页面	1.给杨艳媚搭建calc接口环境 2.拼接home页面的屏幕适配	3h 2h	3h 2h	done	耗时不耗脑，劳力不劳心。
Wed		跑通新页面的计算流程	在新页面上跑通所以流程	1d	1d	done	全部完成，明天进行页面优化。
Thu		发布一个Max的稳定版本	周一的稳定版部署在阿里云主机上	2h	1d	主要问题 1.同账号多人登录问题 2.主机硬盘空间太小，导致测试不了几个系统空间就要满了	Linux 下让进程在后台可靠运行的几种方法 1.nohub 今天为了在服务器上部署max，所以需要我们的一个主机上部署全部的节点，杨总和鹏哥给的命令是“nohub commod &”，开始使用这个命令，所有节点会写入到一个文件中，所以让每个节点进程都重定向到不同的文件，使用“nobub commod >filename 2>&1 & ”重定向进程的流但是出现新的问题，就是最新打开的线程会把之前打开的节点关掉，导致进程也关掉。 2.setsid 后来自己百度，找到了一个新的方式setsid commod，这种方式是提升自己的守护线程（父线程）为系统线程而不是当前终端线程，以此来实现后台运行，但这样做的缺点就是它后台运行后，我调不回前台了，也不能将他的输出重定向到文件。 3.Ctrl+Z

							<p>这种方式就是先暂停程序，然后jobs查看后台运行线程id，fg+id 将线程调回前台来激活线程，然后bg在后台运行，这样做好处是随时可以调回前台，但是缺点也明显，不能重定向，而且需要短暂暂停线程。</p> <p>4.screen</p> <p>这个命令很强大，是运维人员必备的技能，但是需要单独下载，而且杀鸡焉用牛刀，我也不是很会，就不说了。</p> <p>后来的解决方案，说来丢人，之所以我起不来节点是因为之前的主机中就已经有节点启动了，所以我重启了一下主机，使用“nobub commod >filename &”就搞定了。</p>
Fri			1.配合运营对发布的内测版本测试 2.根据鹏哥给的测试dome封装前端和后端的webSocket工具类 3.替换之前所有im写的代码	4h 4h 2h	4h 6h 0h	running	<p>新的Web Socket比之前的环信方便和干净许多，虽然相对少了许多功能，但是对于目前系统完全够用了，而且自己写的代码可以做到那种对于代码随心所欲的控制。</p>
Sat							
Sun			1.根据鹏哥给的测试dome封装前端和后端的webSocket工具类 2.替换之前所有im写的代码 3.对于web socket多终端登录的测试	2h 4h 2h	0.5h 5h 2h	done	<p>周五迷糊了，今天思路清晰很简单就完成了封装，代码拼接也很顺利，但是测试很耗时，而且出现了一下小问题，目前调试通过，已经初步符合产品给的产品原型的需求。</p> <p>而对于多终端的登录问题，也得到了很好的解决，但是对于统一账号多处登录的一些同步和限制问题还是需要考虑和优化的。</p>

Date: 2017 11 第_2_周 11/13-11/19

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	Max 内侧版本优化	完善并测试项目稳定性	1.开通四个市场，并偷摸测试一下 2.写个贼拉全面的部署脚本 3.合并代码，提高入库效率	2h 2h 4h	8h 1h 2h	done	程序员也不应该依赖debug，UnitTest可以让你一遍一遍的重来，但是人生就一次。 今天部署的时候不够细心，只要少修改了一个文件，就要重新编译，打包，上传，解压，启动，在目前公司带宽不足的情况下，尤为耗时，所以这坚定了我早日写脚本的决心，不能依赖工具，但是一定要提高效率。
Tue		发布旧版拼接新版本	1.建立和阿里云主机的rsa公钥连接 2.接着部署线上版本，争取给出一个可用的版本出来	1h 10h	1h 10h	done	早上来发现昨天测试的线上版本出错，目前暂时发布为两个市场进行线上测试。然后，噩梦来了。。。。 11:00-12:00 测试通过，但是环信回调出现问题，并且每个市场入库4条 12:00-13:00 使用张弛的新bson优化效率问题，并且检查代码中所有的回调 13:00-15:30 测试未通过，原因有1.环信传参2.jar冲突3.sbt的jar引入 15:30-18:30 修改jar冲突后，测试，通过，41min可以跑通INF和SPE两个市场 至此，Max2.2正式发布完成。。。。 发布对于使用linux的人来说并不难，但是发布一个可用的版本，并且是在团队协作下发布，如果不规划好，真的是噩梦啊，这要多多像前辈学习，下次不再犯。
Wed		1.在新的服务器上对ASI_R市场测试 2.对sprint3的程序进行测试	1.合并所有分之代码，拼接sprint3 2.发布拼接好的sprint3 3主机测试AI_R市场 3.1 java,scala,sbt安装 3.2 mongo安装 3.3 mongo数据库还原	30min 1h 3h 30min 10min 30min 0min	30min 1h 7h 30min 1h 30min 30min	sprint拼完了 线上AI_R已经开始测试	发现每次发布，时间估计都不准，非常，非常，非常的浪费时间，主要原因还是更改运行环境之后要修改一些配置，但是每次都会遗漏一些，而根本原因就是工程的不熟悉，对代码的不熟悉和对运维的不重视，运维还真的不是轻松活。 下载mongo走了点弯路，但是在鹏哥的指定下轻松解决，还是那句老话，交流是解决问题的最快办法。 最麻烦的还是部署，好的代码规范和好的行为习惯总是可以在不知不觉之间减少许多问题，反之亦然。。。。

			3.4 sbt环境下 下载 3.5 配置新的 运行环境 3.6 测试INF	30min 2h	10min 5h		感觉自己越来越像一名程序员了，但是距离好程序员却越来越远。
Thu		使用云 主机对 所有市 场进行 测试	昨天ai_r没跑 出来，今天测 试其他市场	1d	1d	runnin g	今天测试只有INF市场可以通过，稍大一些的就会出现 问题。后来听说问题解决了，而且效率翻倍，恭喜项目 组的杨总和张驰攻城拔寨，打下一片江山，希望我明天 的项目部署可以顺利理清所有路径，顺利通过。
Fri			继续测	1d	1d		今天除了不断的优化脚本，记录测试，还和张驰互相 交流了一下工作内容和感悟，收货颇丰。
Sat		改bug	1.一些页面样 式和逻辑问题 2.继续测 3.使用play的 filters对页面访 问进行权限限 制	2h 1d 2h	1h 1d 4h	done	和鹏哥研究很久的filters，杨总一看命名就给出了 解决方案，让我震惊羡慕的同时也越挫越勇，编程要 细心，要动脑，不要思维惯性，google不是万能。
Sun			继续测	1d	1d	done	本次对 2017 年 5 月 18 个市场进行测试，通过 12 个，通过率 66.6%。 已通过的市场中，时间最长的 为 AI_S (44min36s)，最短的为 DVP (27s)，其 他大多分布在 5-15min 之间。 未通过的 6 六个市场 失败原因一样，都是系统资源不足，导致各结点之间 短暂失联。 我们已经想好解决方案并正在实现，就是 利用二级缓存实现一种重连机制，可以在节点重启后 回复场景继续计算。

Date: 2017 11 第_3_周 11/20-11/26

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	提高max系统稳定性	1.在max新架构中加入Redis二级缓存 2修改代码逻辑，逐步替换掉FSM	1.修复周六的权限bug 2.对201706的panel市场测试 3.研究calc的FSM，然后逐步干掉	1h 2h 6h	3h 2h 4h	running	今天改bug超时，现在想说，不好好写代码的应该打屁股。 进行测试很浪费时间，写测试报告很练习口才，头重脚重这事情还要杨总教，真是蠢哭了，下次不会了，言简意赅，不说废话。 研究FSM，发现流程不难，代码真的挺复杂，改代码有点像拆炸弹一样，我得好好研究一下。
Tue			1.sprint开始会议 2.阿里云测试 3.了解学习Redis 4.逐渐蚕食FSM	2h 1d 3h 5d	2h 1d 3h 5h	继续改架构	现在头有点乱，逻辑清晰，代码复杂，小心应对，稳扎稳打。
Wed			逐渐蚕食FSM	4d	8h	提取了2个半删除了一个	今天效率还算不错，没浪费前几天的磨刀，但是刚提出filter节点，听说这个功能不要了，哎呀，很心酸啊，以后要及时沟通。
Thu			逐渐蚕食FSM	3d	8h	基本没进展	今天明明一直在工作，但是晚上一回顾感觉毫无进展，明明有思路，有想法却很迷茫，这种感觉很玄妙啊，目前对接到calc，如果为了工作而工作还是可以很快作完的，但是我想在完成任务的同时能够理解杨总的思想，看懂他之前写的代码，顺便重构一下现在的代码，所以可能有些乱，走火入魔了。但是现在接触scala半年了，每每接触杨总关于内存管理的谓词编程部分，还是知其意不知其形为何故，知道目的，但那种写法，那种思想勉强看懂，却难以掌握，哎，没有慧根啊。

Fri				2d	8h	group 模块完成	几乎是一口一口啃的这部分代码（跳过了最精华的杨总的谓词编程），可以说有些本末倒置了，买椟还珠了，但是每天都在尝试去理解一下。 明天开始calc模块，go
Sat							
Sun							

Date: 2017 11 第_4_周 11/27-12/3

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	提高max系统稳定性	1.在max新架构中加入Redis二级缓存 2修改代码逻辑，逐步替换掉FSM	1.完全干掉FSM	1d	1d	顺利	今天在张弛的帮助下，终于干掉所有的FSM,并且杨总也讲解了接下来的架构方向，经过上这一周的工作对于计算这一块也不算陌生了。
Tue			1.重写监听和Master节点路由机制解决节点失联问题 2.学习压力测试工具Gating	2h 6h	张弛完成 6h	完成	在杨总的悉心指导和张弛的艰苦奋斗下，终于解决了在现阶段资源不足而经常出现的节点失联问题，Max新架构的改造也终于告一段落了。
Wed			学习压力测试工具Gating	5d	1d	学习中	通过比较目前比较适合三款压测工具load runner，jmeter和Gating最终选择了以Scala编写的Gating
Thu							
Fri							
Sat							

Sun							

Date: 2017 12 第_1_周 12/04-12/10

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	提高max系统稳定性	测试Max的Client页面能支持的最大用户并发登录和查询的数量	1.发布最新版本(完成失联后重连并恢复场景功能)的Max版本。 2.测试Max系统登录页面的负载能力	1h 7h	20min 7h	done	使用Gating压测工具，本机对本机Max登录页进行压测。在1000用户并发访问的条件下，个别资源未能成功加载的概率为38%，在500用户并发访问的条件下，个别资源未能成功加载的概率为2%。加载失败的资源多为bootstrap文件，失败原因皆为连接资源超时(15s)。按照杨总要求，登录页面在2s内加载完成，且通过率100%的条件下，最大用户并发数为310。
Tue			1.测试线上Max系统登录页面的负载能力 2.测试线上Max系统登录功能的负载能力	1h 3h	1h 未完成	为了测试登录功能，做了一件贼蠢的事，想使用压测工具自动化测试就是找死啊	使用Gating压测工具，本机对阿里云Max登录页进行压测。在310用户并发访问的条件下，个别资源未能成功加载的概率为27%，在80用户并发访问的条件下，个别资源未能成功加载的概率为7%，在40用户并发访问的条件下，加载成功率为100%。加载失败的资源多为bootstrap文件，失败原因皆为连接资源超时(60s)。按照杨总要求，登录页面在2s内加载完成，且通过率100%的条件下，最大用户并发数为12。
Wed		1.和张弛合力分离计算节点，部署在两台机器上 2.学习MongoDB分片技术	1.测试线上Max系统登录功能的负载能力 2.提取算能管理中所有的ip到配置文件 3.完善SCP功能	2h 3h 4h	2h 3h 未完成	未完成	本机对阿里云Max登录功能进行压测。按照杨总要求，在2s内有返回结果，且通过率100%的条件下，最大用户并发数为10。 由于昨天测试AI_R计算未通过，错误信息为大量节点失联造成集群崩溃，今天杨总给出了目前资源条件下的新架构方向，提前分离各个结点，进入分布式时代，将计算和还原节点提出并互斥，这样最耗费资源的结点离开，其他结点的稳定性大大增加，同时提高了系统稳定性。

Thu			1.配置两台云主机环境 2.测试SCP功能	2h 4h	1h 6h	running	接触Max已经很久了，看它就像自己孩子一样，但是每次该代码都能发现他生父生母留下的一些不用的代码，控制自己的强迫症，明年重构。 今天主要消耗时间就是master的功能混肴，它是转发和管理，千万不要进行任何逻辑处理，不然麻烦死了，亲孩子都要要丢掉的冲动。
Fri			1.完成在两台云主机部署Max 2.完善发布脚本 3.学习MongoDB Sharding	2h 2h 4h	6h 2h 0h	云主机和本地都在使用双机测试AI_R	在云上部署的时候，经历两机rsa免密无效（自己好了），mongo不绑定ip（配置文件语法问题），redis不自启动（次优办法解决），scp后无法解压（压缩方式和路径问题）等问题后，完美实现了两机器运行，晚上正在跑着困扰已久的AI_R。因为云主机两个机器一个四核一个八核，所以现在只能写死用四核跑AI_R，否则会出现严重错误，数组越界。
Sat			1.继续云主机双机测试 2.学习MongoDB Sharding	2h 6h	1h 7h	继续学习	云上测试很顺利，一切状态都在计算之内，但是本机测试不顺利，在张弛电脑运行到还原阶段，在写入了2000万条数据之后，MongoDB的写入速度明显变慢，最后就卡主不动，看来分片势在必行，而且我要快点完成就行了。
Sun							

Date: 2017 12 第_2_周 12/11-12/17

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	提高max系统稳定性	使用MongoDB的分片技术提高max数据还原速度	1.继续云主机双机测试 2.学习MongoDB Sharding	2h 6h	1h 7h	继续学习	<p>之前因为云主机测试加入二级缓存的Max新架构在运行AI_R这个超大市场时,还是会出现资源不足问题,造成结点大规模失联的云down现象(此状态不可恢复场景)。</p> <p>上周周为解决节点失联问题,Max架构提前进入分布式时代,在云主机上使用两台机器上部署Max架构,在高性能的机器上部署资源利用率奇高的计算节点和数据还原节点(两节点互斥,不会同时计算和还原),另一台低性能的机器部署Master等其他节点和页面资源,以此减少其他节点的对系统资源抢占,提高计算节点资源利用率,同时提高了系统稳定性。</p> <p>目前两台机器分开部署Max加入二级缓存的新架构版本,对于AI_R这个超大市场,测试四次通过率100%,采用8线程运行时间约为8.5h,稳定性测试通过。</p> <p>云上运行的很顺利,一切状态都在计算之内,但是本机测试就不爽了,在张弛电脑运行到还原阶段,在写入了2000万条数据之后,MongoDB的写入速度明显变慢,最后就卡主不动,看来分片势在必行,而且我要快点完成才行了。</p>
Tue			1.薛总开会讲一下公司的产品,前景,战略,服务,收获量多 2.学习MongoDB Sharding	2h 6h	1h 3h	暂停,拼接版本	<p>开会前的插曲:迟到五分钟客户就不见你了,生意没了,前途就没了,诚信是生意的基本,守时是诚信的基本。</p> <p>关于“水位监控”的插曲:杨总问我啥是水位监控,我说,是三峡水电站看大坝的不,看涨水了,赶紧拉闸放水,然后我就少了一只胳膊。纠正一下,这是常识问题,杨总不会无聊的问我三峡的大坝事情。“水位检测”是运维人员检测服务器网络流量,系统资源的一个统称,通过检测的“水位”可以知道当前系统的运行状态,用户活跃度和系统瓶颈,方便快速改进系统和进行排错,这是我看的一篇介绍服务监控的博客,写的很直接SGM设计与实现</p>
Wed			1.准备和前端的拼接工作 2.测试现版本框架各市场的	2h 8h	1h 未完待续	ok	<p>自己的坑自己填,scp的锅要我要背好。果然着急没有好软件,要稳扎稳打的来。</p>

			运行时间 3.根据杨总的 clean plan， 清理max代码 4.修复SCP一 个bug	5h 1h	1h 4h		
Thu			1.统计现架构 所有市场的计 算时间 2.拼接版本上 线	8h 3h	8h 8h	上线出 了点问 题	版本发布出现问题，开始是websocket连接异常，原因是这次我画蛇添足，所有的地址非要加上80端口（不加默认就是80），导致链接不上，最后在他的创造人钱鹏帮助下解决。 之后就是结果检查在Redis取不到我们想要的数据，原因是和张弛没有进行及时的沟通，发生了一下小插曲。 本来问题都很简单，也就本着自己的事情自己做的态度，就没及时联系他们，结果耽误了整个项目进展，这问题多少次了，还没记性。运维人员必须要有快速定位问题的能力，还要有一个好的电脑和网络。
Fri			跟着郭人豪测 试线上版本	1d	1d	未完成	本来以为顺顺利利的一次发布，结果除去昨天的两个问题，还有一个问题就是计算201704年ELQ市场时split出错，分析原因是panel文件最后有一空行，但是其他文件没有，接着查找原因，是对于同数据合并时，如果发生数据进位时，数据会缺几个字节，这就很可怕了，指针错乱，可能出现很多情况，结尾空行只是其中一种，指针真的太危险了，以后要注意，争取不用，好麻烦。
Sat			跟着郭人豪测 试线上版本	1d	1d	未完成	未知的bug还有很多，感觉好迷茫。
Sun							

Date: 2017 12 第_3_周 12/18-12/24

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	提高max系统稳定性	修复现版本已知的一些bug	1.解决月份解析错误的处理 2.解决web socket会失联的问题	10min 3h	10min 1d	感觉找到问题了，明天来测试	把max的代码过了一遍又一遍，感觉到很多冗余代码，思想上也不统一，所以想抽空清理一下，方便维护。 今天学习了张驰的agent-master架构，错误重算机制，对于actor的几种策略模式可以参考 Actor生命周期理解 学习。
Tue		清洗max的代码，提高维护性，降低以后新人门槛	1.继续配合郭人豪测试max系统 2.一共8个结点，平均2h一个	2h 16h	2h running	项目稳定，清理代码	今天实在受不了自己的强迫症了，调好了bug之后，就赶紧重构了max，先看完了杨总逼着看完的《阿里巴巴java开发手册》，然后根据张驰昨天给我讲解的思想，逐步删除了max中大量不用的代码，删的时候紧张激动，删完感觉真爽啊！！
Wed			1.内部测试版顺利发布 2.清洗calc代码 3.清洗restore代码	1d	1d	在目前阶段，每次版本发布都要控制着几个版本的代码，熟能生巧，今天发布很顺利	曾经的我一直迷茫着三个问题，1.以后要做什么，2.我应该学什么，3.那我现在应该学什么 未来虽然自己做主，但是命运由不得自己安排，所以这些问题应该反过来，我要明白自己在做什么，才知道我要学什么，知道现在学什么就知道接下来应该学什么，当自己所学所会积累到了程度，就知道自己能成为什么。 自己拙见，但是谢谢杨总一直一楼为我们引得路。我不嫌弃自己做的每一份工作，每样工作都有相应的提升，试过才知道合不合适，学过才知道怎么配合，就像LOL，从补刀（写一段好代码）练起，了解各个位置（工程，运维，测试，架构），才能形成配合（好团队，高效率工作），ADC虽然打人疼，可是脆啊，还是需要辅助保护的，不会补兵（规范代码）可不行。等这些搞懂了，才有去调节符文天赋（刨根问底，求甚解）的必要，不然到头一场空。
Thu			1.删除老架构代码，并测试最新代码	1d	1d	感觉没啥难度，其实问题可	今天终于把Max所有代码都整理完了，晚上使用最新代码跑压力测试。 总结 XSS 与 CSRF 两种跨站攻击 : 看了这篇文章，了解了两种跨站攻击的原理和常见

						多了	<p>攻击方式，并结合Max写下一些思考。</p> <p>虽然Max项目没有评论功能，也没有互动功能，用户A想将自己的脚本注入页面，并让用户B访问，不太可能，但是身为一个Web开发人员，不相信每一个get，post请求应该成为一种意识。</p> <p>不过天才总是快人一步，既然你不相信每一个请求了，那黑客就冒充一个真正的请求 (CSRF),这个无耻的黑客技术，目前没有一种可以完全防御的手段，我们能做的就是提高它的难度 (Auto2.0短token)，以上攻击的损失相对来说还可以挽回，还有一个最最最无耻的，就是杨总常说的DDOS攻击，我好好一个程序员，让黑客逼得看了些啥？以后不仅要规范编程，还要安全编程，程序员很辛苦， ㄟ(ˋωˋ)↗</p>
Fri			1.发布本次版本，版本升级 2.继续学习 Sharding	3h 6h	3h 6h	看完书了，开始尝试搭建	<p>这次版本发布的很顺利，测试的也还算顺利。今天首次拿清洗好的代码给产品测试，压力还是有的，不过因为都是自己的代码，改起bug来还是很爽的，不错。</p> <p>书看的有点慢哈！</p>
Sat							
Sun			1.阿里云压测 2.尝试搭建 MongoDB的分片	2d 1d	2d 1d	running	<p>上次一遍的压测没过，原因是以前的资源崩溃问题，我在comeo中用sender返回运行结果，而sender记录的是栈地址，新的实例是不同的，造成了死信，改完就好了。</p> <p>昨天连续3次的压测在第二遍开始没有通过，仔细追踪代码，我去，这弱智的bug，因为每次写入不同用户的panel是相同的，我还用panel做了其他表名！所以程序混乱了。</p> <p>我都不知道该怎么喷自己了，蠢死。</p>

Date: 2017 12 第_4_周 12/25-12/31

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	搭建Spark架构	1.get杨总未来的spark+MongoDB思想核心 2.学会MongoDB的Sharding 3.和张弛搭建出Spark架构	1.本机搭建出MongoDB分片的dome	1d	1d	done	<p>书上教的搭建方式过时了，网上教的方法太新了，我们用的MongoDB版本为3.4.4，分片使用的配置服务器必须是一个副本集，还好这次磨刀时间长，对副本集有所了解，同时没有太过依赖网上和书上教程，才这么顺利的搭建完成。</p> <p>需要注意，在mongos添加分页时，不能使用本机内网ip和localhost，必须使用局域网或外网ip.</p> <p>参考文档： http://www.cnblogs.com/zhoushijinyi/p/4635444.html Mongo学习笔记： http://note.youdao.com/noteshare?id=98933bfb19b7d1a11c46dcccdefe530b </p>
Tue			1.寻找max新代码中的毒瘤 2.完成历史数据导出	1d 3h	未找到 5h	done	<p>昨天的压测又没通过，主要都是在大市场计算时崩溃，但是失联问题明明已经解决了，而且每次测试的原因都不一样，还好有张弛和钱鹏抽出宝贵的时间帮助我，本以为对max有一定能力支撑了，看来还言之过早。</p> <p>第二个任务虽然又比预计的超时了，但是主要原因是对工作量估计不准，这段时间的充电感觉自己的能力还是有提高的。</p>
Wed			1.寻找max新代码中的毒瘤 2.继续测试 3.学习一下Spark	1d 1d 1d	4h 1d 1d	done	<p>昨天晚上连续两次跑一小时以内的14个市场，结果28个job全部通过，之前的问题定位到大市场上。</p> <p>最后终于在张弛夜以继日的不懈努力下，终于帮我找到了代码中存在的一个严重问题，有个巨大的list进行foreach的时候，我在循环内创建redis实例，我嚟，我以为就遍历了8遍，其实会遍历几万、几十万次呢，这内存无极限啊，还好张弛眼神好，找到了这颗毒瘤，今天压测有信心通过。</p> <p>白天对钱鹏拼接的新版本就行业务测试，进展顺利。今晚打算重新挑战17个市场的压测。</p> <p>附7：2017.12.26最新代码的测试结果</p>

Thu			1.编写开机自启mongodb分片集群的脚本 2.配合郭人豪进行流程测试	2h 1d	2h 1d	done	集群的启动顺序是关键，配置服务器必须最先起，最后停。路由服务器最好最后起，最先停。之后很顺利了，毕竟这几天已经蠢得够久了。。。。
Fri							
Sat							
Sun							

Date: 2018 1 第_1_周 1/2-1/7

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon							
Tue	在一个sprint中完善产品部门需要的功能	1.使max支持恩华公司 2.导出功能优化	1.上午对sprint05进行review 2.下午进行sprint06的开始会 3.对max版本升级并测试	3h 5h 1h	3h 5h 20min	done 今晚测试一下新架构对AI_R的计算情况	接触Max2个多月了，今天Jacky在为我们讲解sprint06的方向的时候，才感觉自己真正的认识Max，收货颇丰。 其实项目不是一个人的任务，你不能只了解自己那一块，要互相了解，还要有发散思维，对客户未来需求有一定预测，还有最重要的，沟通是王道，要沟通，要理解。一定要沟通，不要自己认为；理解是目的，不要有太大的主观意识。
Wed			1.修改panel算法，支持单独cpa文件的处理	1d	1d	done 待测	为了应对以后每个公司都有自己的清洗规则，也就有自己的panel算法，所以用工厂模式加上配置文件的反射修改了算法架构，不愧说scala是对java补充和进化，对于java而言那么复杂的反射，scala只需要两句话，语意清晰，代码简练。
Thu			1.完成恩华panel算法的测试 2.进行恩华算法50个jobs压测 3.配合钱鹏进行max整体流程测试	4h 5h 2d	5h 5h 未完	算法完成 明天拼接代码，并测试	还算顺利，不知不觉开始清理client的代码了
Fri			1.解决恩华的上传的源数据	1h	0.5h	继续加入恩华	首先恩华panel算法50job压测通过，平均2min/个，准备拼接发布一个恩华可用的线上版本。

			中title不同问题 2.测试client时，由于早先加入filter后一直没有测试过，出现很多问题，所以整理一下所有接口 3.加入恩华历史数据	2h 2d5h	1.5h 未完		然后就想唠叨一下，今天为了发布支持恩华的版本，所以需要整理数据，添加公司和员工账号，所以钱鹏让我修改一个“小小的”bug,个人信息展示的公司是密文，不合理，要明文显示。之后我就进入了一个逻辑怪圈，代码改的我很是头晕，实在是太难维护了。这种吃力不讨好的工作，还是少做为妙，抓紧时间完成本职工作吧！！
Sat							
Sun							

Date: 2018 1 第_2_周 1/8-1/14

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	在一个sprint中完善产品部门需要的功能	1.使max支持恩华公司 2.导出功能优化	1.拼接版本，加入恩华逻辑。 2.服务器上正式分片 3.加入恩华历史数据	4h 4h 8h	2h 6h 未完	晚上在测试	今天进展顺利，流程已经拼接完成，并且测试通过，今晚将跑恩华16,17两年的数据并存入数据总表，这样Max的两个检查功能就不再是花架子了。对于数据库分片，又产生了一些问题，杨总的目的是将MongoDB变成分布式的，也就是想Facade一样，只保留agent的一部分，这对于大多数情况是可行的了，但是Max有一个功能是bson还原，也就产生一个新的集合，查找各种文档，博客，也没找到合适的解决办法，可以对新添加的集合（名字未知）自动按照规则分片，所有这里需要考虑和研究一下。
Tue			1.修改所有代码中写死公司的代码，并测试恩华流程 2.修改mongoDB默认端口 3.在数据入库之前就对还原的集合进行分片	4h 1h 4h	4h 2h 有点麻烦	running	修改个端口还超时了，其实主要原因是昨天没测试，数据库端口影响的地方还是很多的，配置文件就好几个。 今天发生了一件大事，MongoDB被人黑了，哈哈，干一年活了，终于让我遇到个劫道的了，还好是测试环境，但是对于默认端口，默认配置等安全问题，一定要重视，毕竟现在不是在学校过家家了。最后还剩下一个功能，就是通过code来分片，这个还在研究和淌水中，敬请关注。。。
Wed			1.追查昨天出错原因 2.继续准备历史数据	1d 1d	1d 1d	原因找到了，怎么解决是个问题	本以为已经顺利完成了“构建恩华数据池”的任务，但是今天测试计算流程，在结果检查的时候，发现个别月份没有数据。在数据库中也没有找到相关数据，所以要么是聚合出错，要么是分片出错，通过层层追查发现，果然在分片后，mongo进行均衡时出现错误，但是根据mongo的设计风格，本次均衡出错，会等待下次均衡，并不影响读写。可如果是聚合出错，但是日志没有任何错误信息，而且以前也已经进行了大量测试，相对来说聚合还是比较稳定的。 最后在聚合节点的日志中发现数据库连接断开了，各种日志分析，终于在/var/log/message这个系统日志(新技能get)中发现了明确的报错信息--》内存超出，是个麻烦事，杨总给的方案是给个类似于缓冲的数据库实例，一切操作都在这上面实现，然后在同步到集群，这个方案也许可行，但是浪费资源

							，时间，而且麻烦，对于想偷懒的程序猿来说，我是不想直接尝试这个的。还好利用一天的时间，把恩华30个月的数据跑完了(数据很大可能有些缺失，日后在修复)，结果查询已经可用了，钱鹏也在赶工样本检查，所以说即将有个版本可以使用了，虽然数据不完全正确，但是功能都是有的了。
Thu			1.和郭人豪测试恩华，并验证结果正确性 2.OOM Killer这个杀手不太冷，但是超级烦 3.补充恩华新增的三个药品	1d 3h 3h	1d 4h 1h	done	最近几天的测试都是数据库down掉，而且是被系统down掉，也就是被传说中的杀手OOM Killer(Out of Memory)干掉的，所以今天着重说一下这个杀手。 因为Mongo可以无限申请内存的这个特点，所以它极其容易被OOM Killer盯上，尤其是在建立索引的时候，如果Mongo进程无故突然终止，而日志没有任何记录，就应该检查一下/var/log/messages，查看是否有关于mongod进程的信息。（如果真发生以上情况，多数会有一条内存超出信息）。 当系统没有交换空间，并且可用内存不断减少时，OOM killer就会变得非常敏感（随意杀人），所以最简单的办法，就是加入交换空间，让OOM killer放松下来，如果真杀了进程其实也没关系，无须恢复数据，重启结束的进程即可。
Fri			提高Max稳定性和效率	1d	1d	running	昨天加入了swap后，果然数据库稳定性大大提高，一次没有崩溃了，但是程序加载速度巨慢无比，而且稳定性也不是很好。 主要有以下几点： 1.数据还原bson时，MongoDB连接驱动会掉线，导致还原失败 2.还原时有可能产生actor死信 3.聚合时，MongoDB连接驱动会掉线，导致聚合失败 4.聚合时有可能产生actor死信 5.登录后的首页加载历史数据时间久（2min），甚至有时候会加载不出来，但是程序没报错 以上五点问题，可以归结为一个原因，数据库的效率问题，解决方案正在研究，杨总已经对给出的几个方案做了分析和评价，都不是很满意。
Sat			1解决MongoDB集群出现Query failed with error code -5	4h 2h	1h 15min	done	昨天的前四个问题都是因为查询数据库时，由于现在是分片集群，并且要查询的集合是使用游标进行遍历的，所以driver就会创建分片数量的游标，也就是分片数量的Client实例，占有内存的同时，还可能出现一种叫做“游标超时”的异常，这就是昨天

			and error message的问题 2.解决首页加载过慢问题				的四个问题产生的原因了，但为何会产生这个异常没有自己的追查。 现在产生的一个新的大问题就是结果检查的时候，由于数据池的量太大，导致加载前11个月的数据很慢，甚至有的时候会超时，具体解决办法下周考虑。
Sun							

Date: 2018 1 第_3_周 1/15-1/21

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	在一个sprint中完善产品部门需要的功能	1.解决Max的效率问题 2.学习Spark	1.提高结果检查的加载速度 2.提高保存的稳定性 3.学习MongoDB的Map Reduce，理解其思想	2h 2h 4h	0.5h 0.5h 4h	done	昨天测试的保存还是没办法100%的成功，所以今天想要研究研究，结果只是些微的调整了一下代码，将建立索引和分片的逻辑提到了还原之前，也就是先对一个空表建立索引和分片，之后进行还原，结果大大出乎意料，不仅保存的稳定性提高了，还原的速度也大大提高，在1min内就能还原完成。
Tue			1.配合产品和需求方测试 2.修改页面显示的文案 3.和郭人豪整理明天review的文档	1d 1h 1h	1d 0.5h 1h	done	开始学习spark，年后就要用到了，spark这门思想和Map Reduce有种异曲同工的感觉，不愧是当前最火的大数据框架，很吸引人。
Wed			1.Sprint06Review 2.沟通整理下个Sprint思路 3.学习Spark	2h 2h 4h	2h 2h 4h	done	<p>本次Review中，主要是郭人豪陈述了一些目前项目中的已知问题和未做的功能模块，还有每次添加新公司新产品时，匹配表的更新交互问题。</p> <p>之后时薛总和彭总对于Max的一些期望，和现阶段比较需要完成的一些功能。</p> <p>之后就是李总和杨总对未来Max发展道路上的一些方案，要进行的5步走计划，其中数据仓储是基础，是我们一直想要得到的东西，而半自动化清洗，数据中心则是一个比较麻烦的点，并非一朝一夕可以完成，新架构来提高效率是我一直想要探索的一块，最震撼的类神经网络图的算法修改接口是我一直不敢想的一个点，这个功能的实现简直就是让Max成精了，我感觉我们做的不是工具了，而是在造人，一个可以做工具的人。</p> <p>单独说一下，李总曾经开会多次提到的一个概念==>“B端C化”，以前一直很模糊的一个概念，今天谢谢郭人豪和钱鹏解惑，现在Max的账号体系是在对一个公司而言，在我们（提供方），用户只是一个公司，而实际上，用户明确到公司下的一个员工，有一个员工，就有一个领导，有小领导就有大领导，每个角色的需求铁定不一样，C化的过程就是</p>

							账号体系细化，需求明确+权限机制（个人理解）。
Thu			1.和罗然一起检查结果正确性 2.检查数据库总是丢失数据的原因	1d 1d	1d 1d	都未完成	今天不断的备份还原，备份还原多次数据库，还是无法确定数据库出错的原因，不过还是分片的可能性最大，这个问题先放一放。 然后今天为了和罗然比对结果正确性，实际的使用了一下MapReduce功能，不得不说，很方便，MongoDB的用户体验和产品思想我真的是越来越喜欢了，他把MapReduce虽然是一种查询处理操作，但是它把这次计算的结果存入了一张表中，统一了Map的IO模型，输入和输出都是一张表，并且是一张已经持久化的表，我们可以对这张结果表再次进行任何想要的操作，包括再次MapReduce，以此来将任何复杂的问题颗粒化，就想杨总说的，一个方法做一件事，这样MonogDB就更适合已一个函数式编程的思想来使用了，very good。。
Fri							
Sat							
Sun							

Date: 2018 1 第_4_周 1/22-1/28

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	年前出一个正式可用的Max版本	1. 使Max与线下Max的结果一致 2.Max新增支持安斯泰来	1.和罗然调整max结果的错误数据	5d	1d	未完成	<p>主要想说一个问题，还是老问题，就是对时间的把控上。</p> <p>白天一天没什么进展，临下班的时候，知道我们和线下max差了一个步骤，虽然已经下班，但是还是想早点完成，并且熟悉一下max计算的具体流程，感谢钱鹏为我讲了一遍，并且我结合他给的脑图和代码理清一下，期间不断的质疑他给我讲的和脑图的正确性（计算流程不负责，大数据不在于流程多麻烦，而是数据量），钱鹏都给我一一解惑。</p> <p>但关键问题是这样我还是没能解决现在结果不对的问题，原因就在于，我们不知道他们简简单单的一步“小于0的变成0”到底需要如何实现（维护别人代码真的很难的，再简单的改动也不能掉以轻心），而在这样的前提下，我还非要尝试改动，这是不明智的，不是说加班不明智，我的出发点也是好的，我想完成任务，而做的事情也没很好，我学会理解了max的流程，但是我的期望是不对的，出发点不应该是去改这个bug，或者认为一定可以改了这个问题，因为我只是以改bug为目标，其实在当时这种数据下班的情况下很不明智，还要自己学Excel的使用（还有我这卡死的电脑），还要去揣测他们的这个“改0”问题，完全得不偿失。</p> <p>所以，以后，记住了，出发点和行动统一了，别做无用功，也别浪费时间在一些你不合适的事情（用linux处理巨大的Excel）上，杨总说过，每个方法都是为了解决问题而存在的（当初因为权限问题我推荐使用动态URL解决，但它是为了解决论坛类网站的URL过多和污染存在的），你拿菜刀剪指甲，能用，不麻烦吗？我处理一次Excel用多久？傻不傻。。</p> <p>最后想说，这还算有理有据的说话方式，我是想借鉴一下狼人杀的逻辑性，谢谢Jacky在狼人杀的指导，这种游戏还是蛮锻炼人的。</p>
Tue			1.和罗然调整max结果的错误数据	4d	1d	未完成	<p>需要检讨一下自己，虽然在以前公司有想过去接触和理解业务逻辑，来方便更直观和更好的写出易懂的逻辑代码，但是结果真的不是很理想。</p> <p>而刚来Pharbers的时候，也是这样，而且杨总很好的分离了逻辑和代码，我就更是很少的关注这一部分了，直到这次对数据才开始补习和学起，感谢老郭给我梳理的知识体系。</p>

							同样是计算这部分代码，在本次对数据之前，要是别人问我，计算都干了什么，我就说 (split+group+calc+restore),现在我就会说 (分组，求和，平均，放大，回填)，发现如果了解了整个业务的概念和流程后，很利于读懂代码 (虽然咱们这个流程的好多术语都很接近)，也可以给出相应的补充和优化，同时更方便跟项目之外的人，甚至公司之外的人进行有效的沟通。
Wed			1.检查全部代码，抽象一份新的算法思路 2.和小陆核对算法是否正确 3.和罗然比对数据	4h 2h 4h	4h 1h 3h	未完成	虽然又找到了一个小问题，但还不是造成本次对数据出错的最大问题原因，在小陆的帮助和老郭在旁的辅助补充下，我们找到了一个问题。我就想，为啥我和罗然这几天对数据为什么一点进展没有呢？我看不懂代码吗？罗然不会Excel，看不懂数据吗？都不是，我们是在沟通中，出现了信息丢失，我们丢包了，每次沟通都没发现最重要的点，所以今天我把代码的算法抽象出来了，给她讲解了一遍，希望她能发现问题或者不明确，不保证的地方，以此为突破口。 同时，使用最笨的方法，整理出所有的中间数据和信息，别怕自己麻烦，也别怕麻烦别人，一定要在本周把数对上。功夫不负有心人，铁杵磨成绣花针。
Thu			1.继续核对数据正确性	1day	1day	50% 杨总说，没测的代码就是完成一半，但是我感觉马上成功了	今天一早，小陆告诉了我一个劲爆的消息，spotfire对units进行了向下取整，所以max结果中，可能我们线上的才是对的。得到这个消息，我的心情也不知是喜是忧。不过现在项目终于又有进展了，之后经过一下午的努力，线下去掉了取整这步，中途又修改了几次匹配表，201705INF这块试金石终于对上了。 之后想一鼓作气搞定恩华的，结果赶上匹配表更新，双方时间都很紧，而且换文件，跨系统和沟通这几点又浪费了一些时间，所以拖迟到明天。 嗯，我们离成功又跨了一大步，哈哈哈。 最后，特别表演一下郭人豪同学，是他不断的督促我和鞭策我 [菜刀]，我的进展才可以这么快 (好像不快)，依照我的性格和工作方式可能不会在别人工作的时候去打断人家，所以经常会。。。感谢老郭最近教我的东西，而且他一直提醒我记录心得，完善对接文档，我感觉非常非常有必要，给力的产品。
Fri			1.继续核对数据正确性	4h	4h	done	今天一早，罗然帮我检查了一下恩华17年10月数据，一次通过，之后检查9月，11月均通过，Max

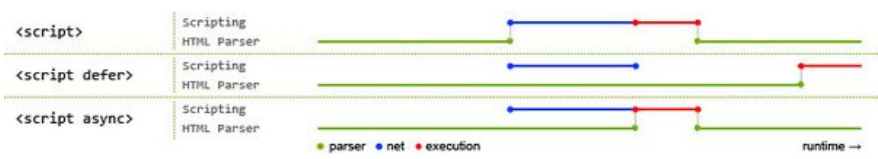
			2.书写Max检验报告 3.整理老郭留给我的三份文档	3h 3h	3h 未完成		正确性检查通过。 以下是根据近期的工作情况整理的三份对接文档： 1.附8：Max对于新增(修改)公司(市场)对接表 2.附9：MAX正确性验证易错参考表 3.附10：MAX最新算法
Sat							
Sun							

Date: 2018 1 第_5_周 1/29-2/4

日期	目标 Objective	关键结果KR	工作项目	时间颗粒预期	完成用时	完成情况	自我回顾 及评价
Mon	年前出一个正式可用的Max版本	1. 配合老郭修复当前Maxbug 2. 书写软件著作权3篇	1.TMIST业务讲解 2. 和罗然检查恩华历史数据 3. 充电，学习医疗大数据理论知识	3h 1h 4h	2.5h 3h 4h	完成	今天和罗然对历史数据，发现其中一个月份对不上，经过测试发现果然是聚合有问题，会使总表数据变少，但影响不大。因为杨总说下一个架构会取消聚合步骤，所以先放一放吧。
Tue							
Wed							
Thu							
Fri							
Sat							
Sun							

附录：

附1：defer和async两种动态加载时序图和区别：



附2：Scrum_Max第一轮黑盒白盒测试结果：

10:00 - 12:00 发现的Bug

1. 注册激活流程
 - a. 输入的input中只能为中文，虽然是中国市场，但应该有公司吊吊的喜欢用用英文吧
 - b. 不响应Enter键，需要手动点击提交
 - c. 页面有水平滚动条
 - d. 《授权码过期》链接无跳转
 - e. 《如何获得授权码链接无跳转》
2. 登录流程
 - a. 不响应Enter键，需要手动点击提交
 - b. 页面有水平滚动条
 - c. 无法通过《获得账号》链接跳转到注册页面
 - d. 无法登录原因不同，如密码不正确，邮箱为注册等原因皆表示为密码错误，原因杨总以解释
3. 找回密码流程
 - a. 提交邮箱后，链接无跳转
 - b. 填写邮箱后，返回的链接和激活账号的邮箱一样，并且功能不能使用

17:00 - 19:00 剩余的Bug

1. token邮件中的token如果太长会导致样式错乱
2. 激活账号页按钮，注册页按钮，找回密码按钮样式
3. 可以多次使用激活账号的邮件来重复重置密码
4. 通过激活账号功能首次登陆Max会在页面上端出现一串token的乱码
5. 找回密码功能没有邮箱格式验证，并且无视填写的邮件，只给指定的账号发送重置密码邮件
6. token填写页面填写token后有回车使会出现“unknow error”异常

附3：Scrum_Max第二轮流程测试结果：

10:00 - 12:00 剩余的Bug

1. 《验证码过期找回》功能没有邮箱，姓名，电话三者的检查
2. 账号激活邮件和密码找回邮件可以重复使用来修改密码，但是有超时时间
3. 密码找回功能非常“智能”，填的错误的邮箱，都能给对的人发送邮件

附4：开发过程中内存优化的几点建议：

在Scala开发过程中，可以通过以下几点规则在一定程度上控制JVM的内存

1. 减少创建的临时对象数量（如 `StringBuilder` > `StringBuffer` > `String`）
2. 减少中间变量，利用Scala的返回值特性可以将多条语句写到一条表达式中
3. 减少代码中`print`，`log`等日志的输出
4. 减少转移到老年代的对象数量（老年代的GC处理非常浪费时间，而对于相对较大的对象实例，在伊甸区创建后会跳过幸存区直接放到老年区）
 - a. 修改JVM，增加伊甸区大小
 - b. 尽量避免创建大容量对象，控制对象大小
5. 减少Full GC执行时间（这个与开发代码无关）
 - a. 如果试图通过消减老年代空间来减少Full GC的执行时间，可能会导致 `OutOfMemoryError` 或者 Full GC执行的次数会增加。
 - b. 如果试图通过增加老年代空间来减少Full GC执行次数，执行时间会增加

附5：copy and swap思想：

它的概念是，通过深复制创建一个本地副本，然后使用 swap 函数来复制数据，并用新数据交换旧数据。临时副本，然后 destructs，取出旧数据。我们留下了新数据的副本。

我理解的可能有些片面，但大致意思应该是没跑。用FP的思想理解就是immutable，结合 `MappedByteBuffer` 就更好理解了，

- 1. 首先打开目标文件 (`new RandomAccessFile(file,"rw")`)
- 2. 打开文件到内存的通道(`getChannel()`)，将文件映射到逻辑内存中 (`map(FileChannel.MapMode,seek,size)`)，
- 3. 加载了一部分buffer到物理内存 (`get()`)
- 4. 对这部分内存进行处理 (`put(byte)`)
- 5. 进行copy and swap，与逻辑内存中的数据进行swap

学自[MappedByteBuffer解析](#)

附6：sprint02中测试bug:

- 1.账号管理中心，左侧tab无法切换
- 2.系统退出功能无效
- 3.账号中心的修改个人信息，页面有大量问题，不祥写了
- 4.修改密码功能对修改条件的验证
- 5.账号中心退回功能
- 6.bd管理中心，对于员工账号的操作工具栏，无效
- 7.正式账号的停用功能未实现
- 8.待激活页面的详情信息不准确
- 9.预约通知已处理，未处理的显示区分
- 10.无法申请授权码的错误处理和展示

附7：2017.12.26最新代码的测试结果

经过上一次的代码清洗以后，测试结果一直不是很稳定，经常出现Disassociated状态，或者出现死信，究其原因，多是因为资源不足，导致某个节点重启，sender返回消息无法收到。修改后，对于计算时间不长(1h以内)的小市场有效，对于大市场，由于失联时间较长，计算数据过大的情况下，还是无法计算成功用了。今天在张弛的帮助下终于找到了问题的关键，有个巨大的list进行foreach的时候，我在循环内创建redis实例，导致实例化对象过多，文件打开次数也巨大，所以出现了现在这么可怕的结果。

好的，自我谴责完了，以下是最新代码的测试结果：

压力测试：昨天去除所有1h以上的市场，并发两次测试，共28个jobs，全部通过，中途有失联和重启现象，但系统都成功处理，但总耗时17h，比预想时间长许多。

流程测试：对加入“保存到数据库”和“导出”功能的Max进行流程测试。

1. 单个市场 (INF):

生成panel	计算	保存到数据库	导出
4min	4min	5min	<1min

2. 三个市场 (INF,SPE,URO)

生成panel	计算	保存到数据库	导出
17	20min	25min	<1min

3. 五个市场 (INF,SPE,URO,AI_W,CNS_Z)

生成panel	计算	保存到数据库	导出
30min	40min	30min	<1min

在计算时，Master节点出现过失联，导致计算丢失了一个job，并且Master稍后重新加入集群，所以没有任何error信息，现在怀疑是akka的http和master串行启动，使master节点没有单独的master节点稳定，明天打算将http和master分开启动，在尝试5个以上市场的流程测试。

附8：Max对于新增(修改)公司(市场)对接表

根据以往Max系统新增或修改公司或市场时，运营团队和工程团队经历了许多困难，为了减少同一困难重复发生的概率，特制此表。

MAX新公司/市场上线对接表				
参与人员	运营组	罗然	对接时间	2018.1.18
	工程组	齐钟昱		
公司(新增或修改)		恩华		
市场(新增或修改)		麻醉市场		
伪代码	对于每个新增市场的panel算法的伪代码	已提供伪代码 注意事项：算法与辉瑞一样，但原文件只需传CPA，并且时间过滤比例为1/5.		
样本检查	测试源文件	已提供2017年1月至11月CPA源文件（title为中文，命名与辉瑞不同）		
	universe_# #_online.xlsx 文件	已提供以线下一致的universe_麻醉市场_online.xlsx文件(去除文件后大量空的单元格)		
	20××年未出版医院名单.xlsx	已提供		
	补充医院.csv	提供以`31`.toChar 为分隔符的csv		
	匹配表.xlsx	已提供匹配表（去除文件后大量空的单元格）		

	通用名市场定义.xlsx	已提供通用名定义 (去除文件后大量空的单元格)
	线下panel	已提供线下panel结果供线上结果验证
	各市场基准值	已提供16年12个月份的基准值
结果检查	线下Max结果	已提供线下Max结果供线上验证
补充		

附9：MAX正确性验证易错参考表

根据上次验证线上Max正确性耗时过久，特制此表帮助双方人员梳理易错点，方便快速排查错误。

MAX正确性验证易错参考表				
参与人员	运营组	罗然	验证时间	2018.1.18
	工程组	齐钟昱		
验证公司		恩华		
验证市场		麻醉市场		
前置条件	保证双方panel完全一样	已通过		
	保证双方使用相同的universe文件计算	已通过		
结果对比	样本医院回填情况	样本医院回填正确		
	非样本医院中与线下Max结果有出入的医院PHA_ID	无		
算法检查	运营检查是否更改过算法	未更改		
	工程检查是否修改过算法逻辑	未修改		
	工程检查是否修改整体架构	未修改		

中间文件	双方检查中间生成的sum和ave	相同
	工程检查生成的中间文件是否符合业务逻辑	已提供线下Max结果供线上验证
补充		

附10：MAX最新算法 (overdue)

1. 按照核心数将panel分成小份
2. 读取对应市场的universe表(25000家医院)，遍历生成计算数据
 - a. 将当前遍历的universe数据和根据(YM+min1)去重的panel进行笛卡尔积，生成计算数据 (panel × universe)
 - b. 计算数据中年份、最小产品单位、市场以panel为准，其他以universe为准，sales和units暂时设为0.0
 - c. 对if_panel_to_use为1 (需要放大) 的数据补充sales和units
 - i. 根据年+月+最小产品单位+PHA编码找到多条panel数据
 - ii. 将找到的panel的sales和units求和，填到计算数据上
 - d. 计算数据生产完成
3. 根据segment+最小产品单位+年月对计算数据分组
4. 计算每个分组的和
 - a. 分组的sales求和记为sum1
 - b. 分组的units求和记为sum2
 - c. 分组的西药收入求和记为sum3
5. 计算每个分组的平均
 - a. 分组的sum1/sum3记为ave1
 - b. 分组的sum2/sum3记为ave2
6. 遍历计算数据，写入最终结果f_sales和f_units
 - a. 对于if_panel_all为1 (样本医院) 的数据，f_sales=sales,f_units=units
 - b. 否则，对于ave1或ave2小于0的数据，f_sales=0,f_units=0
 - c. 否则，f_sales = ave1 * 西药收入 * factor，f_units = ave2 * 西药收入 * factor
7. 过滤掉f_sales=0并且f_units=0的数据，最终的Max后的结果