

齐钟昱		Max_Dev		Scrum - 03	
Okrs	WO 1: 在1.5个月内完成Max内部版本上线 KR1: 在1个sprint完成本地版的MAX版本（除注册流程，包含三公司 <u>计算</u> 流程，登录流程和 <u>查询</u> 流程），计算过程要求安全稳定，单用户使用100%率，查询流程采用杨总要求的二级缓存架构，要求每次query在2s内渲染到页面 KR2: 在2，3个sprint完成内网集群版的MAX版本（包含 <u>注册</u> ，计算，查询等全部流程），MAX系统支持到五个公司（加入拜耳，信立泰） KR3: 在4个sprint使用JsonApi替换原有JsValue的架构，更加贴合微服务思想，应变以后日渐变化的需求				
	不断学习，不断进步，不怕困难怕平凡 SO 1: 2个Sprint做到Scala进阶，学会函数式编程思想 KR1: 掌握杨总经常提起的微服务架构思想，并应用到Max的逻辑层 SO 2: 2个Sprint进行朔本逐源，学习操作系统概念和编译原理 KR1: 我找了一篇操作系统的博客，把每个知识点都写成一片自己的文章 SO 3: 1个Sprint成为“高级程序员”，学会Scala宏注释 KR1: 利用宏注释和微服务思想完成新铁马项目				
OKR文档准则					
<div>I. 检查清单</div> <div>1. 制定的Objective与公司的战略总体目标相吻</div> <div>2. Key Results 符合SMART原则</div> <div>- Specific 具体的</div> <div>- Measurable：可以度量的</div> <div>- Attainable：可以达成的</div> <div>- Relevant：与终极的目标之间有相关性</div> <div>Time-bound：具有明确的时限</div> <div>3. 目标的数量控制在3-5个：少而精，聚焦，找出关键因素，集中优势资源完成</div> <div>4. 部分目标可以是有野心的：激发创新，找到解决问题的非常规方法</div> <div>5. 把目标公开给每一个人：全员公开可以让个体对目标的理解一致，方便协调资源，实现上下同欲。</div> <div>6. 将目标记录下来：随着时间的推移，目标很容易被淡出视野，要不断提醒自己。</div> <div>II. OKR日报规则</div> <div>(1) 1.5个月为一个OKR周期</div> <div>(2) 需每日完成日报（若遇特殊情况，如加班、请假等，可依情况调整）</div> <div>(3) 两周做一次阶段性小结</div> <div>小结：</div> <div>1)整理两周内遇到的问题给出解决方案，帮助大家了解整个项目的情况，避免再次犯同样的错误</div> <div>2)自己整理的可共享学习资料</div> <div>3)为周末成长会准备素材</div> <div>(4) 命名规则：年/月/日-姓名-OKR日报</div> <div>III.注意事项</div> <div>1. 思考是OKR日报的终极目的，拒绝流水账</div> <div>2. 填写时间请控制在30分钟以内</div> <div>3. 出于共享的目的，请用普通人类看得懂的语言进行填写</div> <div>4. 为了方便阅读，请务必使大家一打开就能看到当日最新日报</div> <div>5. 请发pdf格式档案进行共享</div> <div>6. 日报格式與内容，並非强制硬性限定，只需要满足 回顾、思考、共享、透明等目的，可以自行做出相应调整優化。</div> <div>IV.分类标准</div> <div>A.工作事项（如需细分，各部门自行定义）</div> <div>B.问题讨论</div> <div>C.学习</div> <div>D.会议</div>					
工作	讨论	学习	会议	困哪	

	日期/date	2018-08-01			
	遇到的问题				
	服务器下载Docker镜像的时候,太慢了!太慢了!太慢了! 今天一大部分时间都是用来下镜像了,虽然不用我干啥,但是要一直看着,因为他是不是还会出错. 没有精力学习和干别的活了,就把一些可以ACV的代码做完了.				
	思考/ 自我评价				
	关于同时完成多个任务的并行问题上, 人还是不如机器啊.				
		用时	完成情况	工作事项描述	
	工作	8h	running	发布了六个服务, 不包括Routes, 因为他是主要逻辑,改动频繁, 同时发布一次太慢了. 目前登录功能OK了.	
	日后安排	1. Docker 初学 2. 医院,关卡等模块化分离			
	吃饱撑着没事儿闪现的Idea				
	给你个机会提点建议和意见				

	日期/date	2018-07-27			
	遇到的问题				
	思考/ 自我评价				
		用时	完成情况	工作事项描述	
工作	8h	running	提炼 Company，User，和二者关联关系，对于bind的抽象和逻辑。 下班的最后一刻去请教了一下杨总，感觉自己在系统抽象上,还是差了一点, 不过这次就差了一点, 关于ESB直接调用User这一块, 应该也走UserCompany这个小的ESB, 会好一点.		
日后安排	1. Docker 初学 2. TM后台模块化分离 2.1 User 2.2 Company 2.3 Role				
吃饱撑着没事儿闪现的 Idea					
给你个机会提点建议和意见					

日期/date	2018-07-27				
遇到的问题					
今日趟坑无数,幸好杨总铺路!!					
写下几点注意事项, 省的之后的各位重蹈覆辙.					
1. Redis在外网无法连接, 在集群中的连接时, 主机name不能使用别名,也就是”redis”必须是ip才可以.					
2. 杨总之前启动的service千万不要重启,可以更新. 不然太可怕了. 如果,哪一天真的重启了所有service,尤其是三个mongo,那么一定要找杨总修复,或者,连接mongo,然后输入					
> rs.initiate({_id:"rs0", members:[{_id:1, host: "primary:27017"}, {_id:2, host: "secondary:27017"}, {_id:3, host: "arbiter:27017"}]})					
3. 每次发布都真的很浪费时间,所以自己写好单元测试, 再发布, 切记! 切记! 切记!					
思考/ 自我评价					
不看理论直接做, 是不是盲人摸象!					
	用时	完成情况	工作事项描述		
学习	2h	running	学习Docker, 现在只会使用, 有些模糊概念, 配置完全不懂, 还需充充电!!		
会议	30min	done	杨总每天例行检查和任务分配		
工作	5h	done	单独提出一个Auth已经成功发布到阿里云了. 没有重大改动就不需要动了. 提出了一个公共模块, pharbers-client-base, 以后提出的每个module引入就可以实现常用功能. 写了一个简单的routes-module, 用来做初版的ESB, 但是距离杨总的设想还很早. 以上所有的代码都在新工程pharbers-micro-service中, 开发请下载,并重建分支.		
日后安排	1. Docker 初学 2. TM后台模块化分离 2.1 User 2.2 Company 2.3 Role				
吃饱撑着没事儿闪现的Idea					
给你个机会提点建议和意见					

日期/date	2018-07-26		
遇到的问题			
<p>Docker完全不会, 准备今晚,一天速成.</p> <p>拆解了一个Auth node, 这里问题不大, 但是想将一些通用的功能(比如杨总的CURD 宏, formatJson 之类的提取出来, 试过 play, sbt mvn 三种打包方式) 都有些麻烦, MVN是环境, 因为通用的部分是play的很多功能, play 打包更麻烦, 没有插件, sbt打包, 太不友好了, 主要是对目录层级的严格要求 , 还有 sbt没有install, 要使用 publishlocal 发布到本地仓库才能引入成功.</p>			
思考/ 自我评价			
	用时	完成情况	工作事项描述
工作	6h	running	节点分离, 概念梳理, 不多说了.
会议	2h	running	杨总给我们培训Docker,并且梳理了任务进度.
日后安排	1. Docker 初学 2. TM后台模块化分离 (完成了), 明天测试		
吃饱撑着没事儿闪现的Idea	最近写了一篇Git入门教程,欢迎指正 https://www.cnblogs.com/clockq/p/9330546.html		
给你个机会提点建议和意见			

日期/date	2018-07-25		
遇到的问题			
<p>学习CPS遇到点问题, 各位帮帮忙!</p> <p>我定义如下方法:</p> <pre>function fact(n,ret) { if (n ==0) ret(1); else fact(n-1, function (t0) { ret(n * t0) }); }</pre> <p>调用:</p> <pre>fact (5, function (n) { console.log(n); // 在Firebug中输出120 })</pre> <p>据说这是一种尾递归优化, 好在哪了呢?</p>			
思考/ 自我评价			
<p>王垠的40行代码,究竟diao在哪里?</p> <p>https://www.cnblogs.com/clockq/p/9368524.html#cps</p>			
	用时	完成情况	工作事项描述
学习	6h	running	一直在学习, 东西在博客
会议	2h	running	tm sprint 01 会议
日后安排	1. Docker 初学 2. TM后台模块化分离		
吃饱撑着没事儿闪现的Idea	最近写了一篇Git入门教程,欢迎指正 https://www.cnblogs.com/clockq/p/9330546.html		
给你个机会提点建议和意见			

	日期/date	2018-07-12	
	遇到的问题		
	<p>今天在拼接TM的报告接口的时候, 发现如果我有了R跑出来的基础数据, 可以在logic层进行一些统计(页面显示需要).</p> <p>本来想着这样做(在不读多余表的情况下),可以对数据库进行优化, 同时减少R的代码逻辑.</p> <p>当然,我写在这里,一定是又想错了,被杨总教训了一顿. 主要错误如下:</p> <p>1. 首先,我这样做是减少了数据库的复杂度,但是减少真的有减少R的代码逻辑吗?(好好想想)</p> <p>2. 在一个Alpha的版本中, 首要目的只有一个,就是能用,实现原则也很简单(灵活,随动,代码少).</p> <p>对于Alpha阶段的代码, 要做到的主要就是:</p> <p>1. 能用</p> <p>2. 能扩展</p> <p>3. 扩展简单</p>		
	思考/ 自我评价		
	<p>编程就像画画, 不要想着写书法!</p> <p>2-8 定律的应用场景, 我又收集了一种.</p>		
	用时	完成情况	工作事项描述
工作	8h	running	1. 了解 TM 的报告页逻辑, 进行数据库设计和接口的实现
日后安排	1. TM 报告页的数据库设计和接口实现.		
吃饱撑着没事儿闪现的Idea	<p>1. Alpha</p> <p>Alpha版本仍然需要测试, 其功能亦未完善, 因为它是整个软件发布周期中的第一个阶段, 所以它的名称是“Alpha”。Alpha版本通常会送到开发软件的组织或某群体中的软件测试者作内部测试。</p> <p>2. Beta</p> <p>Beta版本是软件最早对外公开的软件版本, 由公众参与测试。一般来说, Beta包含所有功能, 但可能有一些已知问题和较轻微的程序错误 (BUG) 。</p> <p>3. Release Candidate</p> <p>Release Candidate (简称RC) 指可能成为最终产品的候选版本, 如果未出现问题则可发布成为正式版本。在此阶段的产品通常包含所有功能、或接近完整, 亦不会出现严重问题。</p>		
给你个机会提点建议和意见			

日期/date	2018-07-11		
遇到的问题			
数据库设计上, 以前的原则太简单了,满足三范式, 字段尽量少. 拜托,那是80年前的思想了. MongoDB给我都白瞎了.			
思考/ 自我评价			
产品逻辑和程序逻辑, 根本不是双胞胎!			
	用时	完成情况	工作事项描述
工作	8h	running	<div>1. 最近一周实现了actions表的抽象,和通用接口的实现, 然后最后面加了一层Format(由鹏哥来写的), 以此来适配森哥想要的数据格式.<div>1.1. 用递归实现了一个通用的 <code>DBObject => Map[String, JsValue]</code>的函数,存在一个Bug,就是数据库中的List,里面的对象必须是Object,也就是<code>{}</code>, 可以修复,但是代码就难看了,😁,所以在此注明,反正不是大问题,哈?</div></div>
日后安排	1. TM报告页的数据库设计和接口实现.		
吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见			

	日期/date	2018-07-05	
	遇到的问题		
	<p>黄鑫的R在调用一下api，如group by的时候，报错信息会提示本地的一个packages.tar找不到，什么原因目前并不知道，杨总看了之后，认为是在R的驱动层（代码中）设置了SparkHome导致的，还认为应该将一些代码和jar包发送到Spark机器中，而我认为，R是数学分析的一种工具，更偏向于应用，应该不会走代码中涉及更多的环境问题吧，好像IaaS和PaaS一样（可能不是很恰当）。</p>		
	思考/ 自我评价		
	<p>今天力气活偏多，晚上有时间思考一下。</p>		
		用时	完成情况
			工作事项描述
	学习	4h	done
	工作	1h	running
	工作	1h	done
	日后安排		
	吃饱撑着没事儿闪现的Idea		
	给你个机会提点建议和意见		

日期/date	2018-07-04		
遇到的问题			
黄鑫安琪使用R调用Spark集群有问题，搞了很久不行啊，这跨语言跨系统的问题果然啥时候都是个问题。			
思考/ 自我评价			
	用时	完成情况	工作事项描述
工作	3h	done	和森哥鹏哥设计TM前后通讯协议
工作	2h	done	硬编码方式解决森哥前端假数据问题
工作	1h	done	杨总給梳理了一下TM的数据库设计思路，已经运行表的具体细节，明朗许多
日后安排	1. 数据库设计，主要以下三个部分 1.1 用户权限scope设计 1.2 实体类（医院，产品，销售人员）关系设计 1.3 运行表action设计，已经状态更新问题		
吃饱撑着没事儿闪现的Idea	路上和老铁聊起博客，感觉他自己建立的这个站真的很不错，定制性很强很强，可以自己实现评论，目录，样式。以前认为他对用户不友好，现在才明白以前jackey和杨总经常说的，站在用户角度，不要自己以为自己以为的，臆想用户需求。擦的，我都程序员了，还怕对用户不友好，我们是什么人，只要你东西好使，能用，别人行我就行，所以github没有浪费时间把gitpage做成大众化的论坛。至于它SEO做的不是很好，嗯，确实不咋好。		
给你个机会提点建议和意见			

	日期/date	2018-07-03	
	遇到的问题		
	<div>1. 今天自己动手,写了一个宏注解,在鹏哥的帮助下,调理了两个小时,终于成功启动了这个Dome,同时对Scala的宏和编译的过程也有了新的认识.</div> <div>写宏的时候,感觉自己就像回到了没有IDE的那个年代,不能依赖IDE,不然真的编译不过,要按照最最最基础的知识,调整编译的顺序和过程,同时也发现自己真的是太依赖工具了,忽略了一些本质的东西.</div>		
	思考/ 自我评价		
	<div>哎,百感交集.我的”宏儿子”还想茁壮成长呢,我对他抱有多大期望啊,都可以想象他的出现对于这个世界来说意味着什么,简直是惊天动地,可惜,他”流产”了.</div>		
		用时	完成情况
			工作事项描述
	学习	3h	running
			自己编写宏注解,测试通过 最近的学习成果, 有兴趣的可以看看我的笔记 微服务架构1 https://www.cnblogs.com/clockq/p/9240596.html 微服务架构2 https://www.cnblogs.com/clockq/p/9251890.html 微服务架构3 https://www.cnblogs.com/clockq/p/9256235.html
	工作	3h	done
			1. 使用泛型完成了Controller层的隐式调用
	日后安排	1. 先暂时放弃宏实现的微服务架构, 使用Max的后端框架,重写TM新项目	
	吃饱撑着没事儿闪现的Idea		
	给你个机会提点建议和意见		

	日期/date	2018-07-02		
	遇到的问题			
	1. 我自己写的注解和宏都不好使,我们用的真是一种语言吗?			
	思考/ 自我评价			
	现在很着急,真的很着急!			
		用时	完成情况	工作事项描述
	学习	6h	running	1. 学习杨总写的Json-api工程,中Controller层的学习 1.1 主要是JsonApi的解析 1.2 宏的编写和调用 1.3 杨总神乎其神的隐式用法 今天的学习成果, 有兴趣的可以看看我的笔记 https://www.cnblogs.com/clockq/p/9256235.html
	日后安排	1. 学习杨总写的Json-api工程, 深入理解表达式树和新的语法糖 1.1 重点是宏的编写和解析 1.2 表达式树的概念要深入		
	吃饱撑着没事儿闪现的Idea			
	给你个机会提点建议和意见			

	日期/date	2018-07-01		
	遇到的问题			
	表达式树,懵的一B			
	思考/ 自我评价			
	1. 元编程的概念,本来很模糊, 杨总醍醐灌顶,没必要咬文嚼字,这只是种思想.			
		用时	完成情况	工作事项描述
	学习	6h	running	1. 学习杨总写的Json-api工程, 深入理解表达式树和新的语法糖 1.1 Scala的宏 1.2 隐式宏 1.3 宏注解 今天的学习成果, 有兴趣的可以看看我的笔记 https://www.cnblogs.com/clockq/p/9251890.html
	日后安排	1. 学习杨总写的Json-api工程, 深入理解表达式树和新的语法糖 1.1 重点是宏的编写和解析 1.2 表达式树的概念要深入		
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见				

	日期/date	2018-06-28		
	遇到的问题			
	思考/ 自我评价			
		用时	完成情况	工作事项描述
	学习	6h	running	<div>1. 学习杨总写的Json-api工程, 深入理解表达式树和新的语法糖</div> <div>1.1 package object</div> <div>1.2 scala的自定义注解</div> <div>1.3 结合注解简单理解表达式树</div> <div>1.4 implicit的”高级”用法</div> <div>今天的学习成果, 有兴趣的可以看看我的笔记</div> <div>https://www.cnblogs.com/clockq/p/9240596.html</div>
	日后安排	1. 学习杨总写的Json-api工程, 深入理解表达式树和新的语法糖		
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见				

	日期/date	2018-06-20	
	遇到的问题		
	<div>1. 依旧出现jar缺失的小问题，搞定</div> <div>2. OOM了，回忆最近修改的代码，最大的嫌疑就是数据库连接实例太多了，连接池改成单例的。解决</div> <div>3. 测试astellas的时候，记得有两个市场数据不对，就反复查了几次max结果，突然client就崩溃了，猜测是数据库连接超时，而超时的原因是数据库实例已经到了最大的连接数，重启client验证一遍，在查询样本结果时，mongodb连接数每次加2，只加不减，查询max结果时，首次连接数加72，之后每次查询加68，这也太扯了，分分钟就满了啊，原因是啥还不知道，把searchFacade单例注入到play中，问题解决</div>		
	思考/ 自我评价		
	用时	完成情况	工作事项描述
学习	3h	running	<div>1. 学习JsonApi</div> <div>2. 继续我的博客园文化</div>
工作	5h	done	<div>1. 集群测试Max出现了一些问题，明天接着看看</div>
日后安排	<div>1. 继续学习JsonApi，理解微服务思想，二者合一，来适应TM的架构需求</div>		
吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见			

	日期/date	2018-06-19	
	遇到的问题		
	<div>1. 还记的前段时间，部署集群时候，一直出现的RPC超时问题吗？今天又出现了，本机开启spark也不好用，但是，我把wifi关掉之后就好了。。。好了(´·ω·`)</div> <div>2. 今天抽空偷偷的测了一下集群的Max，哎，又出了点问题。问题不大，找不到包，找不到类，环境问题，还算容易解决。</div>		
	思考/ 自我评价		
	<div>王垠：“每次仇恨一个东西，你就失去了向它学习的机会。”</div> <div>1. 要让商业逻辑和代码逻辑完美的分开，不然改起来累死人</div> <div>2. 在抽象时，参数类型不同可以尝试泛型，其他用方法传参。泛型是编译期多态，传参是运行时多态。</div>		
		用时	完成情况
	学习	3h	running <div>1. 学习JsonApi</div> <div>2. 继续我的博客园文化</div>
	工作	3h	done <div>1. 更新集群的代码，把最新的代码部署上去，出现了各种各样的小问题</div> <div>1.1 RPC超时的问题，顺利解决了，这次发现贼重要，原因一定是wifi的问题</div> <div>1.2 找不到包，找不到类等环境问题</div>
	工作	2h	done <div>1. 集群测试Max出现了一些问题，明天接着看看</div>
	日后安排	1. 继续学习JsonApi，理解微服务思想，二者合一，来适应TM的架构需求	
	吃饱撑着没事儿闪现的Idea	<div>1. 今天很有意思，学习JsonApi的时候，翻到了一篇博客园的帖子，然后想起来我曾经的博客，就回去看了一哈，博客园有个设置功能，是可以嵌入自定义css/js代码，也就是可以根据博客园官方设定的接口和规范，可以随意的定制自己的博客。这不就是杨总经常提起的PaaS吗？我们提供平台和规范，客户提供算法就可以随意DIY。。。</div> <div>2. 我发现个事哈！地铁的那个手扶电梯不是可以调整方向的吗？而且有的楼梯，只有一侧的手扶电梯，另一侧是楼梯，为啥不根据上下班的时间和大多数人的流动走向，动态的控制这个电梯是上行还是下行呢？因为今天我换乘的时候，很多人都堵在了楼梯那，而电梯只有星星零零的几个人。提供更好的服务，服务更多的人，才是一个好电梯！！</div>	
	给你个机会提点建议和意见		

	日期/date	2018-06-13		
	遇到的问题			
	1. 代码写的还是太烂了，扩展起来费劲，维护起来费劲，修改起来费死劲了。。。。			
	思考/ 自我评价			
		用时	完成情况	工作事项描述
	工作	7h	done	1. 移植MarketTable到MongoDB 1.1 这次改动太大，首先使用了MongoDB后，使用了新的数据结构，带有嵌套结构 1.2 对MarketTable有最大依赖的那个“牛逼”的Builder全部重写了 1.3 对于一些通用的查询，如全部公司，公司名称，子公司，公司下的全部市场等接口重写设计 1.4 老铁的运维中心接口重新设计 1.5 有间接引用关系的小崔的单元测试，微调了一下，使用新架构，并且有更高的控制配置文件的权限
日后安排	1. 对Market的update操作 2. 继续学习JsonApi，理解微服务思想，二者合一，来适应TM的架构需求			
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见				

	日期/date	2018-06-12		
	遇到的问题			
	<div>1. 对于数据库的设计上，还有很大的进步空间，数据库设计肯定是要以最合适（易维护，易拓展）的为由，合理选择和抛弃范式化和扁平结构化。 2. 以前写的代码扩展性是个严重问题，易读只是符合开闭原则之后的一点锦上添花，而绝非重中之重。</div>			
	思考/ 自我评价			
	对于增量数据和query的condition字段不应放在嵌套中			
		用时	完成情况	工作事项描述
	工作	3h	done	1. 修复Excel中末尾为空，则数据缺失的Bug
	学习	2h	running	1. 结合JsonApi的思想理解杨总写的Dome
	工作	3h	done	1. 将MarketTable由Json改到MongoDB中进行管理
	日后安排	<div>1. 填坑，MarketTable的移植问题 1.1 设计新的MongoDB结构，并将MarketTable移植过去，同时兼容老铁的运维中心和小崔的对数测试 1.2 优化一下新的Builder，减少MongoDB的连接次数和每次的内存使用 2. 继续学习JsonApi，理解微服务思想，二者合一，来适应TM的架构需求</div>		
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见				

	日期/date	2018-06-11		
	遇到的问题			
	思考/ 自我评价			
		用时	完成情况	工作事项描述
	工作	3h	done	1. 新增“角色”这个概念，用来对用户的权限和行为进行管理
	工作	3h	done	1. 将MarketTable由Json改到MongoDB中进行管理
	工作	1h	done	1. 帮温总激活office2016，那个新电脑太卡了，而且激活失败，我重新装了一个office2013
	日后安排			
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见				

	日期/date	2018-06-05		
	遇到的问题			
	今天给老崔看代码的时候，发现了以前写的bug，虽然当前版本没影响使用，但终究是肉中刺。 测试力度不够啊！！			
	思考/ 自我评价			
	Jacky和老郭总是能对很多问题进行归类和划分，让复杂的不相干的一系列问题变得简单有条理。以前我一直感觉规则的推行很复杂也很困难，在Jacky的梳理下，把问题分为两类，人工做的，机器做的。人工做的主要就是人工要怎么做（title规范），和不能怎么做（末尾空单元格，内容后大量空行）。之后继续向下细分，一个二叉树就浮现在了脑中。 而之后讨论的关于“修改了算法或者其他计算条件下，对于往期数据是否回滚，又如何回滚”这个问题陷入了无限循环，其实我感觉我的反向推导方式还是蛮不错的。。。。			
		用时	完成情况	工作事项描述
	工作	3h	done	1. 发现可以使用MVN来进行Gatling测试，不需要下载安装 新建了一个gatling的项目，并测试通过，方便之后人员的使用
	工作	1h	done	1. 提取了max模块中paction的可重用部分，测试通过
	工作	3h	done	1. 搭建Gatling测试框架 2. 完成Gatling培训文档
	讨论	3h	done	1. Jacky带领大家整理老郭一直想要推行的规范
日后安排	1. 使用Gatling对目前的Max Play Api进行接口压测，并梳理一套规范 1.1 搭建测试框架 1.2 对Max Play Api 压测 1.3 根据本次压测结果（生产的压测报告）， 结合自己经验，写一个易入手的培训手册 1.4 完成Pharbers Http Api 压测规范 2. 编写部署文档			
吃饱撑着没事儿闪现的Idea	我赶脚，代码review还是比较适合在会议室里安安静静的进行，我们一起把所有的代码都过一遍，当做是一次培训了。咋样，杨帅哥			
给你个机会提点建议和意见				

	日期/date	2018-06-04		
	遇到的问题			
	1. 测试了一下集群，主要问题如下： 1.1 历史记录模块可能是引入方式的问题，现在完全不好使 1.2 今天真正的交给罗然测试了，对于输入文件（源文件），产生错误的可能很大 1.3 对于所有None值和DataSet中的Empty的处理要整理一下			
	思考/ 自我评价			
	事务的ACID特性： 原子性(Atomic) -> 一致性(Consistency) -> 独立性(Isolation) -> 持久性(Durability)			
		用时	完成情况	工作事项描述
	工作	3h	done	1. 上周发布成功了，测试过程中，发现所有的写入文件操作都失败，之前在Mac失败，现在集群失败，老铁的Linux还是好使的，很头疼。 找不同呗，现在的主要区别就是系统不同和用户不同了，集群的用户名都是spark，通过我大量的测试发现，真的是用户名的问题，集群的连接会在NFS机上映射为一个特定用户，而其他机器连接会映射为匿名用户，两者权限不一样，同时发现linux系统连接出来的匿名用户有组写入权限，而Mac的没有，很复杂，很麻烦，总之最后的解决办法很简单粗暴，所有的连接我都强制伪装成spark:spark，也就是NFS服务器的一个管理员，问题解决。。。
	工作	2h	done	1. 鹏哥已经将Ember和Play整合好了，所以我只是发布项目，并且写了两个脚本，没遇到什么问题
	工作	1h	done	1. 我想让部署变得简单点，但是还是需要之后的人有些部署的基础和经验
	工作	3h	done	1. 测试了一下集群，主要问题如下： 1.1 历史记录模块可能是引入方式的问题，现在完全不好使 1.2 今天真正的交给罗然测试了，对于输入文件（源文件），产生错误的可能很大 1.3 对于所有None值和DataSet中的Empty的处理要整理一下
	日后安排	1. 使用Gatling对目前的Max Play Api进行接口压测，并梳理一套规范 1.1 搭建测试框架 1.2 对Max Play Api 压测 1.3 根据本次压测结果（生产的压测报告），结合自己经验，写一个易入手的培训手册 1.4 完成Pharbers Http Api 压测规范		
	吃饱撑着没事儿闪现的Idea			
	给你个机会提点建议和意见			

日期/date	2018-06-01		
遇到的问题			
今天帮黄鑫看看他们的R语言可以使用我们的Spark集群吗，Window肯定是不行了，可是按照我昨天的逻辑，他的Mac我也没成功，完了，昨天的努力好像被不明力量推翻了。			
思考/ 自我评价			
	用时	完成情况	工作事项描述
工作	3h	done	1. play环境部署完成，并测试通过 2. 帮黄鑫连接Spark，未成功
工作	5h	done	1. 发布上一版的MAX到集群上，本来想使用MVN的插件进行打包，后来发现想实现定制化的功能很麻烦，在杨总的开导下选择自己写shell脚本解决。
明日安排			
吃饱撑着没事儿闪现的Idea	想对系统有绝对的控制权，还是要少用哪些别人写好的东西啊！！ 这句话要好好捂捂，好好悟悟		
给你个机会提点建议和意见			

日期/date	2017-05-31		
遇到的问题			
思考/ 自我评价			
	用时	完成情况	工作事项描述
学习	1h	done	1. 学习Spark集群的配置
会议	2h	done	1. Sprint02-05开始会议，分配每人工作事项，并了解杨总讲解的微服务架构
工作	3h	done	1. 对Cent OS重新分区 2. 分区后rsa免密无效了，原来CentOS对每个文件夹会设置一个标签，有兴趣可以自己去看看，不难但是很麻烦 3. Spark开机自启（失败了，太麻烦了）
工作	5h	done	1. Spark集群成功了，杨总希望driver机可以随意调用，我的mac不行，鹏哥的mac不行，老铁的Cent OS却可以，最后发现在driver调用spark集群后，spark会回连driver一个本地接口，这个接口干啥的我不知道，但是在本地开启spark后就好使了。 2. 上面问题解决后，spark写文件时还是出错，因为我们使用的是nfs不是hdfs，可能不是那么好的兼容spark，同时发现nfs文件的所属用户在每台机器和每种系统下还不一样，有点懵。最后修改nfs服务端的配置，不检查subtree权限，并且一切连接已匿名用户存在，解决问题。nfs解决的启发贴 https://www.cnblogs.com/mchina/archive/2013/01/03/2840040.html 3. 想实现Spark的Job并行，失败了
明日安排	1. 搭建四机环境（估计明天也干不完）		
吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见			

日期/date	2017-05-30		
遇到的问题			
在6.5版本时用的是iptables，但是在7版本改为了firewalld作为防火墙，一定要记住			
思考/ 自我评价			
	用时	完成情况	工作事项描述
工作	1h	done	1. 装完一台电脑的系统啦
工作	2h	done	1. mac和Cent OS实现RSA免密还是很麻烦的，浪费我两个小时
工作	3h	done	1. 在【MaxLogic】机上，安装开发环境，具体步骤如下： 1.1 安装java -> scala -> sbt 1.2 安装node.js, -> npm -> ember -> VPN -> 开始下包ing 1.3 安装redis 1.4 安装mongodb 1.5 试运行play，有点问题
工作	1h	done	1. 新机器来了，装系统，配ip
工作	3h	done	1. 对三台机器 【smaster】，[sworker1]，[sworker2]搭建环境，在smaster上搭建NFS Cent OS用不明白，防火墙关了两个小时才关上，在6.5版本时用的是iptables，但是在7版本改为了firewalld作为防火墙，一定要记住
明日安排	1. 搭建四机环境（估计明天也干不完）		
吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见			

日期/date	2017-05-29		
遇到的问题			
<p>Spark 比 Akka 快的原理我真不会吗？ md，会你不说，领导光环的压制效果还是太强了啊！！</p>			
思考/ 自我评价			
<p>每个人的路都是不一样的，我怎么能抽象一条自己的路，留待后人走呢？鹏哥写好了ember的培训手册和学习计划，老郭可以梳理一个sprint的工作成功和所有问题要点，老铁做事总能一针见血，而我一直天马星空，随心所欲，不会制定计划，不会梳理要点，我真能教会别人吗？</p>			
	用时	完成情况	工作事项描述
工作	3h	done	1. 给新同学安装系统，配置环境和软件
Review	1.5h	done	Scrum02 / Sprint03 Review 1. 不要花太多精力在历史查询上，因为即将展示为图表样式，并进行聚合处理，可节省大量资源 2. 样本检查的基准线更新问题以线下为准，有些许误差可以接受 3. 和数据组进行协作后，各种文档的合作和规范的推行要抓紧提上日程
工作	1h	done	1. 对于即将到来的四机时代，设计架构，并做准备工作。
工作	4h	done	1. 制作centos镜像，我妥协了，用centos，可是mac和Ubuntu制作镜像太麻烦太慢了
明日安排	1. 搭建四机环境（估计一天干不完）		
吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见			

	日期/date	2017-05-28		
遇到的问题				
思考/ 自我评价				
<p>今天写单元测试的时候，想实现一个功能，根据一些简单的业务逻辑来给DataFrame增加一列，这一列的值就是之前这个简单业务逻辑（也就是这个函数）的返回值，由此发现了Spark的一个牛逼功能 => UDF(User Defined Function).</p> <p>UDF实现的功能就不多说了，但是Spark提供的这个接口，说白了就是给用户一个开放的平台，连Spark这样的产品或者说平台，都不敢说他提供的内部函数可以实现用户的所有功能，可见，一个封闭的平台是无法拥抱变化的。</p> <p>无论算法也好，工具类还是逻辑都好，最好的重用单位自然是：函数，而不应该是OO里的类。这也是FP流行起来的一个原因，微服务架构也就更加适合我们的Max系统，Max作为一款想向PaaS发展的产品，给用户更大的扩展，更是必不可少的。</p>				
	用时	完成情况	工作事项描述	
讨论	2h	done	找罗然分析Nhwa的panel基准线不一致的问题，结果是出现误差是每月的源文件有更新。	
测试	1h	done	1. 和钱鹏整体流程测试	
工作	3h	done	1. 修改“对数测试”模块的算法和架构	
明日安排	<p>1. 正式开发V3.0.1</p> <p>2. Sprint Review</p> <p>本次Sprint的主要改动有哪些？</p> <p>1. 在程序上和架构上，最大的改动就是引入了Spark和Kafka这两大主流框架，用来进行大数据的处理运算和逻辑层运算层的解耦。</p> <p>2. 业务上，我们已经可以支持恩华，安斯泰来的八个市场，辉瑞的二十三个市场</p> <p>3. 效率上，使用Spark框架后，小市场效率提升13倍，而对于大的市场尤其是超大市场提升更为显著，达到60多倍</p> <p>4. 同时，Max系统健壮性大大提升，不仅节省更多计算机资源，同时可以计算之前很难完成的市场，如AI_R</p>			
吃饱撑着没事儿闪现的Idea	<p>今天才知道一个非常震惊的消息，Spark2.0之后的底层通信竟然不用Akka了，而是使用了一个叫Netty的东西，why？在网上抄了一段答案。。</p> <p>1. 很多Spark用户也使用Akka，但是由于Akka不同版本之间无法互相通信，这就要求用户必须使用跟Spark完全一样的Akka版本，导致用户无法升级Akka。</p> <p>2. Spark的Akka配置是针对Spark自身来调优的，可能跟用户自己代码中的Akka配置冲突。</p> <p>3. Spark用的Akka特性很少，这部分特性很容易自己实现。同时，这部分代码量相比Akka来说少很多，debug比较容易。如果遇到什么bug，也可以自己马上fix，不需要等Akka上游发布新版本。而且，Spark升级Akka本身又因为第一点会强制要求用户升级他们使用的Akka，对于某些用户来说是不现实的。</p> <p>感觉最主要的就是第一点了，版本冲突一直是我当程序猿以来最不愿意面对的问题。</p>			
给你个机会提点建议和意见				

日期/date	2017-05-24		
遇到的问题			
<p>JVM的类加载过程为： 加载 -> 验证 -> 准备(静态变量定义，静态常量赋值) -> 解析 -> 初始化(静态变量赋值，实例变量定义赋值) -> 使用 -> 卸载</p>			
思考/ 自我评价			
<p>Ember.js有点意思，结构分明，我很喜欢，但是对于没接触过或者没有认知的人还是很难接受的。</p>			
	用时	完成情况	工作事项描述
讨论	1h	done	和大家整理这个Sprint任务完成情况
学习	2h	done	1. 学习JVM中的类加载器相关知识 2. 学习Ember.js，争取能看懂鹏哥代码
工作	3h	done	1. 补全三家公司的基准线数据
明日安排	1. 信心是建立在多次测试之上的		
吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见			

	日期/date	2017-05-22		
	遇到的问题			
	1. 测试安斯泰来的过程中，前端进度条无回调，已解决 2. 安斯泰来的米开民市场数据错误，我赶脚可能是我昨天的Universi文件改错了，先忽略 3. 对于安斯泰来的最后一页还是会出现OOM，先放弃			
	思考/ 自我评价			
		用时	完成情况	工作事项描述
	工作	6h	done	1. 后端同时跑安斯泰来的八个市场，经常出现内存不足，提高本机的swap大小，问题解决 1. 和老郭测试安斯泰来的流程
	工作	3h	done	1. 添加辉瑞公司剩余的两种panel算法
	明日安排	1. 继续增加辉瑞公司 2. 整体测试		
	吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见				

	日期/date	2017-05-21		
	遇到的问题			
	思考/ 自我评价			
		用时	完成情况	工作事项描述
	工作	2h	done	1. 添加astellas公司，并进行测试
	工作	4h	done	1. 由于Universe文件的原因，测试的八个市场，除了阿洛克以外均不能通过，所以按照格式修改其余七个文件，并测试，现已全部通过
	工作	3h	1/3	1. 添加辉瑞公司的一种panel算法
	明日安排	1. 继续增加辉瑞公司 2. 整体测试		
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见				

	日期/date	2017-05-18	
遇到的问题			
对查询模块进行的重构和优化，那个RddIndex真的是麻烦的要命，不友好，不友好。但目前为止没找到什么很好的替代方案，先这个吧，太浪费时间了，这次应该不能OOM了，对于我的小黑机来说			
思考/ 自我评价			
Spark的G1 GC，和JVM的传统GC完全不同，G1将Heap分为了若干小区，每个区域内部不再进行新旧分区，这样当创建对象时，直接存放到某个块中，满了就向另一个新块中存，minor GC时，只清理这些新的块，并把存活的对象拷贝到一个块中，当没有资源分配新的块后就会触发full GC。而在标记存活对象时，G1使用RememberSet的概念，将每个分区外指向分区内的引用记录在该分区的RememberSet中，避免了对整个Heap的扫描，使得各个分区的GC更加独立。可以看出G1 GC大大提高了触发Full GC时的Heap占用率，同时也使得Minor GC的暂停时间更加可控，对于内存较大的环境非常友好。			
	用时	完成情况	工作事项描述
工作	4h	done	1. 继续整体流程测试，主要针对查询的OOM问题
讨论	3h	done	1. 和老铁考虑查询的流程和规范
工作	4h	done	1. 重构优化查询模块
明日安排			
吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见			

	日期/date	2017-05-17		
	遇到的问题			
	主要问题有： 1. 首页搜索Count可能为空，后台对Count的缓存处理方式 2. Redis状态太乱 3. 结果查询的地理信息列表，每次都会增多，原因是Redis存放的数据结构为List，并且每次没清空 4. 历史查询的查询条件，前后端出了点小问题 5. 整体测试中，多次查询历史信息，会导致OOM，但后台压测没问题，原因不明			
	思考/ 自我评价			
	1. 在长达两小时的拷问中，重新梳理了一下历史数据缓存池的架构了思路。 2. 任何数据都要有结构，即便是Redis中的也一样，使用以前的递归式查找（uid -> rid -> tid），而不应该图方便而使用硬编码编程（uid + rid -> tid）。 3. 因为业务要求，会agg历史数据，求出一些值，本想将这些信息以类似解决基准线的方法和形式处理，但被杨总否决，不能为了实现业务而开发没有任何抽象的代码。 4. 对于缓存，学问很深			
		用时	完成情况	工作事项描述
	工作	4h	done	1. 和老铁解决昨天的Count缓存问题，因此引发了一系列的思考，Redis状态太乱了，梳理了一下所有的Redis数据，顺便和老铁学习了一下杨总教给老铁的search无敌大内存单缓存数据池。
	工作	3h	done	1. 和钱鹏整体测试，发现OOM问题，原因Spark虽然最大可用内存4g，但计算机无法分配4g内存给Spark，杨总怀疑有内存泄漏可能，正在排查
	讨论	2h	done	就今天的一系列问题，我们三人把杨总拖进了会议室，并对他进行了长达两个小时的拷问
	明日安排	1. 寻找代码中的彩蛋，细节优化		
	吃饱撑着没事儿闪现的Idea			
	给你个机会提点建议和意见	一个理想的程序员应该做到的六个点： 1. 专注眼下 2. 思考力、应变力与推动力 3. Never Say No，没有需求是完成不了的，也不是什么都要去完成的 4. 投资未来，关注社区 5. 善用工具，Google和Script是必不可少的 6. 管理时间，工作学习合理分配 出处： https://www.runoob.com/w3cnote/a-programmer-insight.html		

	日期/date	2017-05-16		
	遇到的问题			
	程序上的问题都很好的解决了，主要是sbt的依赖问题和spark的环境问题，困扰我和老铁许久，不难，现以解决。但我要想个好办法一劳永逸的解决这些问题，不然以后部署还是自己遭殃，因为要替换的文件太TM多了。			
	思考/ 自我评价			
	今天杨总在我脑海里规划了一份华丽的蓝图，以后的法伯将不再只是一个数据分析公司，而是一个平台，因为我们不希望为每个公司编写一组算法（当然，给钱另算），同时我们应该给用户更大的自由，一个好软件应该不是想着怎么绑定用户，而是怎么服务用户，让他觉得没你不可。作为一个PaaS，我想最重要的用户体验应该就是友好的使用方式了，以后的客户如果可以二次开发，那很大的受众应该可以归为程序员行列，而我作为一个程序员，就喜欢调用干净整洁的api。以目前的架构和我目前水平是不可能实现的，因祸得福的，微服务架构也因此走入我的眼帘微服务架构是一项在云中部署应用和服务的新技术。大部分围绕微服务的争论都集中在容器或其他技术是否能很好的实施微服务，红帽也说API应该是重点。见明知意，微服务，体现在“微”这个字上，什么微呢，答案是any，所有细节都应该做到抽象再抽象的最细粒度，因为服务粒度越细，就越能够灵活地降低变化和负载所带来的影响。微服务和以前接触MVC，SSH或者六边形架构都有很大不同，不说服务的粒度粗细，在思想上也是天差地别。如果非要比，感觉和没有ESB的SOA很接近。优势如此诱人，劣势肯定不会少，不列举了。而没有ESB的SOA如何实现呢？FP就起了至关重要的作用，递归式的构建和运行方式加上合理的函数抽象（我也没get到）就可以实现。最后总结一下，不论什么架构，什么技术，什么模型，最最最后返璞归真就是程序员领域的八字真言，封装_继承_抽象_多态，真是年纪越大，看这四个词就越是难以理解啊。附上本段学习博客 http://			
		用时	完成情况	工作事项描述
	工作	2h	done	1. 和张弛对样本检查和结果检查接口。
	讨论	1h	done	杨总查看了我牛逼哄哄的Builder，并讲述了一下未来Paas下的微服务架构，整体理解，细思极恐，感觉实现起来无从下手
	工作	2h	done	1. 和钱鹏整体调试MAX流程，一切模拟线上真实场景，进展顺利
	工作	3h	done	1. 和张弛调试历史查询功能， 这部分比较麻烦，因为目前kafka的一些问题，所以临时以引包的方式调用Driver的所有查询功能，其他接口没问题，但历史查询是一个Spark任务，需要Driver的一些环境，而且logic层是sbt搭建的，所以下载jar包和jar包冲突这块烦的一逼。大部分时间都浪费在这上了。
	明日安排	1. 寻找代码中的彩蛋，细节优化		
	吃饱撑着没事儿闪现的Idea			
	给你个机会提点建议和意见			

	日期/date	2017-05-15		
	遇到的问题			
	测试max时候，spark老是出错，查询sparkUI的日志可以轻易发现是找不到class，解决办法可以按个尝试			
	1. 本工程打包试一下			
	2. 再spark环境中加入相关的包			
	3. 有多个版本的包导致加载冲突，删掉一个			
	现在的公司添加和用户添加流程加入和环信的关联，而删除流程没处理，所以测试过程中，可能有各种小问题，以后会修复。			
	思考/ 自我评价			
	今天写代码，写着写着，我突然就顿悟了，对FP和OO有了一层新的认识，这感觉就想野马挣脱了缰绳，恶狗吃到了骨头（😋），太爽了。			
	OO是什么思想就不说了，干扰读者。就FP而言，定义的每个方法def，更准确的说是每个函数，都是在描述解决问题的步骤，或者可以说是数据的变化，我要一个把数据从什么变成什么的函数。def(jv: JsValue)(func: JsValue => Map[String, Any])，如果这么一抽象，以自身而下的方式（我不知道可以这么说不？）思考问题，我靠，没有难题啊。			
	用这种思想写了一个超级牛逼的Builder（我认为的），等着杨总明天过目。			
		用时	完成情况	工作事项描述
	工作	4h	done	1. 和钱鹏测试Max整体流程，都是小问题，无伤大雅。
工作	1h	done	1. 其实没咋对，各干个的，但是就是不断的map，groupby，reduce。。。。。	
工作	1h	running	1. 和张弛对接的时候，计算总是出错（不是报错，我写的环境好像Spark的错误不返回了），白忙乎好一会，打入一些jar到spark环境解决问题。	
工作	3h	done	1.完成了一个最新版的Builder	
明日安排	1. 整体测试			
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见	今个代码写的太爽了，一填24h不够用，网管再给我充十块钱。			

	日期/date	2017-05-14		
	遇到的问题			
	思考/ 自我评价			
	今天真是让杨总小看了，kafka实现异步调用，自己想的方案都被pass了，杨总讲的很明白了也没搞定，脑海里连个模型都没有，还一直认为这是不能实现的，真是丢了程序员这种爱探索的职业的脸啊，我们认为世界上没有实现不了的东西，只是能力问题。			
		用时	完成情况	工作事项描述
	工作	4h	done	1. 和钱鹏对查询接口 主要检查我接口的入参，和对钱鹏的返回数据是否可以顺利渲染
	工作	3h	done	1. 和张弛对接口 为了达到更快的渲染速度，我们牺牲了计算的时间，其实就是在计算之后，进行了一系列的Map动作，把之后可能会用到的数据先计算出来，并且存入Redis中，在真正查询的时候，取出即可
	工作	3h	running	1. 本来想的流程是driver层，使用环信返回渲染数据，但钱鹏和杨总要求同步查询，所以对kafka进行一些处理，其实具体实现一直很模糊，一直没搞定，所以先使用直接调用的方式先解决，以后再考虑项目扩展应该怎么处理
	明日安排	1. 整体测试		
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见				

	日期/date	2017-05-13		
	遇到的问题			
	思考/ 自我评价			
		用时	完成情况	工作事项描述
	工作	5h	done	1. 更新老铁的代码，重构Max流程和Max的Builder
	明日安排	1. 去除MongoDB中无用的job状态，并将状态管理移入Redis中（优先级高，为查询流程做准备） 2. 利用新Builder编写Search流程，并和老张，老钱调试成功，希望一天搞定(优先级高，开发类)		
	吃饱撑着没事儿闪现的Idea			
给你个机会提点建议和意见				

	日期/date	2017-05-11		
	遇到的问题			
	思考/ 自我评价			
	1. 晚上吃饭，见识了党员的觉悟			
		用时	完成情况	工作事项描述
	工作	5h	done	1. 重构logic层代码，优化所有接口，对前端屏蔽实体类的bind动作 2. 加入Redis管理job状态
	讨论	1h	done	和老铁，钱鹏讨论查询的几个主要接口
	改Bug	3h	done	kafka突然不好使了，原因不知道，重启电脑有效
	明日安排	1. 去除MongoDB中无用的job状态，并将状态管理移入Redis中（优先级高，为查询流程做准备） 2. 利用新Builder编写Search流程，并和老张，老钱调试成功，希望一天搞定(优先级高，开发类)		
吃饱撑着没事儿闪现的Idea				
给你个机会提点建议和意见				

	日期/date	2017-05-10		
	遇到的问题			
	思考/ 自我评价			
	<p>1. Builder是建造模式的一种，主要是避免构造函数中有多个参数时，传参顺序无法得到保证，多一个少一个的也不知道，所以为了向调用者暴露构造细节，出现了Builder模式。但在Scala中这个优势已经可以被传名参数功能取代，但是，对开发者而言也是不够友好。而我们项目现在需要的是根据公司和市场执行特定实例的同一方法，唯一不同的只是构造所需要的参数不同，如同筛选硬币的小盒子一样，可以根据大小落到指定的盒子中，再利用外部文件如（Excel，Xml）加上工厂，就可以实现一个完美调用过程了。哈哈哈哈哈</p>			
		用时	完成情况	工作事项描述
	培训会	3h	done	拜耳的Fancy培训我们医药行业销售的整体概念
	工作	5h	done	<p>1. 完善Builder支持Panel生成</p> <p>2. 对于同时生成多个panel的spark进度合成</p> <p>3. 对所有持久化操作进行压缩保存</p> <p>4. panel执行结束串联一步redis的保存动作，方便老张之后的search</p>
	明日安排	<p>1. 新增一类MergeStepResult方式，提高串行化代码复用性（优先级低，重构类）</p> <p>2. 去除MongoDB中无用的job状态，并将状态管理移入Redis中（优先级高，为查询流程做准备）</p> <p>3. 利用新Builder编写Search流程，并和老张，老钱调试成功，希望一天搞定(优先级高，开发类)</p>		
	吃饱撑着没事儿闪现的 Idea	<p>1. 今天是我来公司一年整，回忆过往，宛如昨天。</p>		
给你个机会提点建议和意见	<p>对于今天这个三个小时的会议，我真的有必要捧场吗？</p> <p>天书一样的内容，浪费了我宝贵的三个小时</p>			

日期/date		2017-05-09	
遇到的问题			
<div>1. 使用Actor对进度条监听去除了线程锁，但因为之前driver层没用到Actor，所以以后的测试要加入Actor环境，所以会很麻烦。@老崔</div> <div>2. 环信目前无法发送信息，可能是昨晚压测过程中没有注释掉，导致发送的流量超出，明天再试试，不行的话再注册一个账号。</div> <div>3. 写的应急Builder不是我想要的，杨总也给了之后的一些指导，有机会再优化吧</div>			
思考/ 自我评价			
<div>结对编程的效率问题？</div> <div>这是网上对于结对编程的解读和优劣分析https://coolshell.cn/articles/16.html，结对编程是上个年代比较流行的开发方式，两人或多人合作开发一个功能，可以有效解决单人的技术短板，遇到技术难点也可以进行快速沟通，毕竟之前两人都做的一份工作，而且是实时同步的，几乎在之前就完成了所需的沟通。同时也可以减少Review成本，Review接下来讨论。</div> <div>但今天不是讨论它的优点，而是要说一说它的缺点，结对编程虽然概念上是有个人为主，有个人为次，但有一个前提条件就是两人的能力要接近。今天杨总说了一句话“我就不喜欢结对编程，这就是1+1<2,不，是1+1<1的一共工作”，对于1+1<2我是认可的，毕竟两个人做一件事，就整个项目而已，进展变慢，但就个人而已，有了更多的学习进步，但1+1<1，我想应该与我们的工作方式有关吧，一个人就可以完成的任务，多个人帮忙不</div>			
分类	用时	完成情况	工作事项描述
站立会	10min	done	个人讲述昨天的任务，遇到的难点，今天的任务
讨论	20min	done	和张弛钱鹏设计job状态的存储方式和存储流程
工作	5h	done	<div>1. 使用Actor来对进度条去锁（优先级中，优化类）</div> <div>2. 使用Actor对环信接口封装，实现异步调用，利用Actor特性增加系统健壮性（优先级高，优化类，虽然目前没问题，但对于以后系统扩展有限制）</div> <div>3. spark数据持久化后gzip压缩（也就是panel文件和max结果文件的压缩）</div> <div>4. 后端Builder的初版完成</div>
改Bug	3h	running	环信无法发送消息，没有任何错误提示，正好刚改完环信代码，以为程序坏了，结果环信可能被限流了
明日安排	<div>1. 新增一类MergeStepResult方式，提高串行化代码复用性（优先级低，重构类）</div> <div>2. 去除MongoDB中无用的job状态，并将状态管理移入Redis中（优先级高，为查询流程做准备）</div> <div>3. 利用Builder重构Max流程（优先级高，重构类）</div> <div>4. 对于所有的持久化操作进行资源管理，并加入压缩参数（优先级高，优化类）</div>		
吃饱撑着没事儿闪现的Idea	1		
给你个机会提点建议和意见	<div>代码Review的机制？</div> <div>代码Review是很有必要的，检查的要点我认为主要有以下几处（摘自杨总发的阿里巴巴Java开发手册）</div> <div>1. 最基本的，变量方法的命名，缩进，包的位置结构，要见名知意，见包位置明白调用范围，重要性</div> <div>2. 其次注释，尽量使用英文，防止跨平台，跨系统时候的乱码问题，对于方法注释，未完成的标TODO，有Bug的标FIX，等等</div> <div>3. 对于历史代码（垃圾代码），比如加入新功能，或者为了适应当前环境临时加入的方法要进行标注，并及时清理，老代码是通过Git管理的，而不是注释</div> <div>4. 最后，代码逻辑，抽象层次，这个我自己就是个问题，知道这部分很重要，但思想这东西还是需要时间积累，像杨总学习</div> <div>Review的方式我建议几种吧！</div> <div>1. IDEA有自己的代码检查插件，但是我们使用过，也不知道功能可以做到怎样，等着各位探索讨论</div> <div>2. 同事之间互检，不仅是一个上帝视角的检查过程，也是一个学习过程，看代码很重要</div>		

	日期/date	2017-05-08		
	遇到的问题			
	思考/ 自我评价			
	分类	用时	完成情况	工作事项描述
	会议	1h	done	sprint会议任务分发15min，之后杨总详细讲解关于查询流程，缓存的一些技术细节和技术难点
	讨论	1h	done	因为提供的辉瑞比对文件和预期不一致（哎，有了规范还能出错），所以和小崔寻找辉瑞在近期测试的方式，因为辉瑞存在的特殊产品（**苏）导致程序出现为捕获的异常
	工作	6h	70%	和钱鹏跑通计算流程，并校准进度条，同时逐渐减少代码的硬编码，改为传参
	明日安排	1. 使用Actor来对进度条去锁（优先级中，优化类） 2. 使用Actor对环信接口封装，实现异步调用，利用Actor特性增加系统健壮性（优先级高，优化类，虽然目前没问题，但对于以后系统扩展有限制） 3. 新增一类MergeStepResult方式，提高串行化代码复用性（优先级低，重构类） 4. 去除MongoDB中无用的job状态，并将状态管理移入Redis中（优先级高，为查询流程做准备）		
吃饱撑着没事儿闪现的Idea	1			
给你个机会提点建议和意见	1			

	日期/date	2017-05-07		
	遇到的问题			
	情景重现： 我：杨总，我们终于把进度条对上了。 杨总：嗯，终于对上了。 我：要不是我为了1%准确跳动，我直接这么写早完事了。 杨总：对于进度条，你这是不可能的，除非是下载那种。 我：是啊，我也想明白了。 杨总：为啥我认为那么简单的事情，你们总是会绕上一圈呢。			
	思考/ 自我评价			
	进度条这个模块我从上周三开始研究，算上周末两天和今天一天写了六天了，才写了一个初版，究其原因有以下几点： 1. 对技术的掌握不足，对于spark中的application，job，stage，task没有一个详细的概念和理解，学习浪费了一些时间。 2. 对自己的盲目自信，想着写出1%的完美回调，却忽略了现有技术，接口和框架限制。 3. 开始的框架选择错误，一味只想追求高难度的完美操作，结果写出了不伦不类代码，之后退而求其次的写了个阉割版，要知道欲速则不达，还有循序渐进的道理。			
	分类	用时	完成情况	工作事项描述
	讨论	1h	done	和杨总，小崔讨论Builder的写法
	工作	6h	70%	完成spark job的进度条回调，并使用IM和前端完成进度条通讯
	明日安排	1. 结合Actor完成进度条的错误处理 2. 和前端核对所有的错误提示		
	吃饱撑着没事儿闪现的Idea	1		
	给你个机会提点建议和意见	1		