



Broadview®  
www.broadview.com.cn

京东研发体系●著  
JD.COM

技术解密



电子工业出版社  
ING HOUSE OF ELECTRONICS INDUSTRY  
//www.phei.com.cn

京东研发体系●著

# JD.京东.COM

# 技术解密

电子工业出版社  
Publishing House of Electronics Industry  
北京·BEIJING

## 内 容 简 介

京东高速的增长、闪电响应的供应链、庞大的团队规模等背后内幕，对于业界一直像谜一样神秘。随着成为中国 B2C 领导厂商以及在纳斯达克上市，京东越来越需要开放自己，与业界形成更好的交流与融合。《京东技术解密》的面世，就是京东技术团队首次向业界集体亮相。本书用翔实的内容为读者逐一解答——如何用技术支撑网站的综合竞争实力，如何把握技术革新的时间点，如何应对各种棘手问题及压力，如何在网站高速运转的情况下进行系统升级等备受关注的关键词题。

本书从 618 大促销、产品演进、技术演进、创新激发、牛人专家五个侧面详细描述了京东研发团队的发展，本书不仅适合于技术人群，同样适合电商平台和其他 IT、互联网行业的从业者。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

## 图书在版编目（CIP）数据

京东技术解密 / 京东研发体系著. —北京：电子工业出版社，2015.1

ISBN 978-7-121-24691-3

I. ①京… II. ①京… III. ①电子商务—商业企业管理—经验—中国 IV. ①F724.6

中国版本图书馆 CIP 数据核字(2014)第 256613 号

策划编辑：张春雨

责任编辑：徐津平

印 刷：北京天来印务有限公司

装 订：北京天来印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：720×1000 1/16 印张：20.75 字数：395 千字

版 次：2015 年 1 月第 1 版

印 次：2015 年 1 月第 1 次印刷

印 数：11000 册 定价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线：(010) 88258888。

# 编委会

特邀顾问：李大学 马松 曹鹏 肖军 何刚 熊宇红  
刘轶 孙晖 陈伟 赵延涛 翁志 江川

总 编：吕建伟

副 总 编：刘艳光 樊晨 刘坤苗 马沛 刘家瑞

编委会成员：许立龙 张侃 李刚 王先科 王大泳 王晓钟  
彭青 王宇 章爱文 暴丹 王晓琼 李昂  
王彪 欧阳波 谭悦娴 陈鲲 李鹏涛 吴新红  
王茹 张华 刘磊 谭晷 王超 杨洪涛  
南婕 王远 李鑫 章耿 丁俊 桂创华  
王松林 梁秋实 赵海峰 郭理靖 孙学斌 侯超  
李松峰 赵刚 靳简明 侯会满 张斯聪 王威  
陈玉兰 高慧 刘朋飞 张立军 胡浩 谢蔚  
马斗 张国利 郭建华 董浩 李帅 张蔷



# 推荐序一

我曾经在很多场合谈到京东可以高速发展到今天的规模的原因，其中最核心的是坚持“倒三角”战略：建立出色的团队；打造财务、物流和技术三大核心系统；降低成本、提升效率；为用户带来最佳体验。在《京东技术解密》一书中，大家会看到技术驱动的力量，这也是京东第一次全面展示自己的技术体系。

过去十年，京东在技术领域做了些什么呢？2004 年刚刚上线时，我们叫京东多媒体。当时京东只有一组简单展示 IT 产品的网页，而如今京东已经是覆盖 PC 端、移动端，并完整支撑物流、订单、交易、商品、财务系统的巨大电商平台；当年我们的技术部门只是一个四五人的小组，现在京东集团有了超过 4000 人的技术研发团队，其中还有不少牛人；最初京东的技术平台是外购的简单 ASP“个人网站系统”，而目前我们的技术架构除了满足京东自身应用需求外，更是具备了为用户提供资源服务的云计算、大数据平台。十年来，京东的技术平台实现了高速增长，成为驱动京东成长的基础，也为京东未来的发展储备了充足的动力。

在京东技术平台的发展过程中，有过很多激动人心的时刻。读到这本书，我仿佛又回到那些对京东成长至关重要的时间点，回想起并肩战斗，现在已经成为京东各个领域骨干的伙伴们。在这里，我要再次感谢京东的技术团队，感谢你们的辛勤努力让京东有了今天的地位，让用户有了优质的体验，让京东的“触角”得以自如延伸。

京东人从来不沉浸在过去的喜悦中，京东的梦想绝不止于此。我们对自己不满足，因为我们坚信京东还有更远大、更宏伟的目标，还可以做得更好。“永不知足、永无止境”的精神流淌在我们的血液里、刻在我们的骨子上。京东的技术平台也会一路披荆斩棘、勇往直前。

技术的成长支撑了过去十年京东的高速发展，放眼未来，技术将不再只是幕后英雄、支撑平台，而将成为驱动京东进一步做大做强的动力来源。按照我们的

规划，京东将以技术为驱动建立先进的供应链服务、开放服务业务和数据金融业务，在我们引以为傲的业务布局上全面发力。

对于用户来说，京东技术平台的进步会给用户带来完全不同的购物体验。我认为，未来用户将拥有身临其境的神奇购物体验，挑选产品及下单会更加自然和便捷；配送则会比今天更迅捷，从而一步步实现京东的愿景——让生活变得简单快乐。

我期待京东技术的美好明天——拥有出色的团队、坚实的基础、务实的架构和创新的管理，让技术驱动京东走得更快、更远。

京东集团创始人兼首席执行官

刘强东

2014.10

# 推荐序二

京东的朋友送来一个书稿，关于京东技术团队十年来走过的路，向我约一个序言。

作为一个消费者，我很喜欢京东网购正品和快速这两大特点。京东仅用十年便快速崛起，成为千万消费者喜爱的电商品牌。要将一个大型服务平台做到优质已非易事，做一个可信任可依赖、服务范围覆盖全国的电商生态，更是无比艰难。通过本书，读者可以感受到京东技术团队十年来的付出和坚持。

## “十年如一日”的坚持

在业务高速发展的背后，技术上会有很多坑很多歧路，即使有一个坑解决不当或迟缓，都可能导致服务质量变得糟糕、业务受损、挫伤品牌。在每一个关键时刻，及时填平这些坑，是技术团队经常要面对的挑战。

应对这些挑战并不容易，除了技能和经验，还需要热爱这个行业及团队。对技术架构的追求是永无止境的，不存在最完美的架构，技术团队需要有“十年磨一剑”的精神，用“十年如一日”的坚持和专注，解决领域内一个又一个的实践问题——先扛住、再优化、再提炼，跌倒了再爬起来。

技术人员往往比较沉静，他们很少面对聚光灯，不会像资本、市场、产品那样有很多盛大的发布会，但他们是真正的幕后英雄。

## 信任和传承的团队文化

大型服务平台高速发展的背后，一定有许多许多的组织问题，少不了磕磕碰碰、问题矛盾。正如没有完美的技术架构一样，也不会有绝对完美的技术组织。技术团队的组织演进也是一个动态发展的过程。只要团队秉持相互信任和主动传承的理念，方法总比困难多。

技术团队的工作很辛苦，但又充满了乐趣和成就感。当一个技术团队在高难度的挑战中，不断战胜自我，磨合出信任和传承，其所处环境就可以成为人才辈出的乐土。

在这里祝福京东的技术团队，可以发展得越来越好，培养出越来越多的技术大牛，研发出更多更优美的技术，给亿万消费者创造更优雅的电商体验。

腾讯创始人之一、前 CTO

Tony 张志东

2014.10

# 打造最有影响力的技术团队

李大学

2008 年我加入京东，见证了京东技术团队从 30 至 4000 人的成长历程，也见证了京东是怎么从业务为主的公司，变成技术驱动的公司。下面，结合京东具体的事来讲一下其中的见闻和感触。

## 京东研发组织架构

京东技术团队目前有超过 4000 名员工，拥有北京、上海、深圳、成都、沈阳 5 大研究院。

京东的系统比较复杂，涵盖电商全流程全价值链，从前端的交易系统到供应链，到仓储、配送、客服以及售后，除了财务报表之外，几乎全部系统都是自己研发的。京东的全部业务都在线上运行。京东 CEO 刘强东现在一年时间里只有一两个月在国内，大部分时间在国外。他通过信息系统可以看到所有的生产情况，所有的数据一目了然，比如说仓储某个环节拥堵了他都知道。

京东的信息系统支持了京东连续多年以高于行业平均增速的速度增长——2013 年市场交易额突破 1000 亿元，达到 1255 亿元。10 年时间，实现了 10000 倍的增长，背后靠的就是信息系统。

在 2013 年年中，京东以“电子商务”为基础的“四架马车”的战略已经基本完善，包括电商业务、物流平台、技术平台和互联网金融业务。在围绕管理提升、组织能力和组织协同方面，京东也正在进行更多有益的探索和尝试。京东正在发生一个蜕变，从一个中国型的电子商务公司，变成产业链的整合者。而京东战略的核心，还是要以技术为驱动。

## 组织架构调整：研发体系分为 9 大模块 4 个平台

管理 4000 人的团队，我觉得组织架构优化是比较重要的一件事。以前京东的技术团队一直是职能化结构：产品部、研发部、测试部以及运维部门。而当我们的产品线和项目多了以后，人也多了，要分优先级就非常复杂，沟通也要非常多，部门协作就很困难。

京东今年对技术团队做了调整，研发体系分为 9 大模块，基本是上按照客户、事业部的组织形式来划分的。

其中，云平台和运维两个部门主要是负责底层的网络服务，以及解决顶层的技术架构问题，是提供技术支持的平台。同时负责研究统一的研发工具，包括日志、监控和流程引擎等，通过这些工具可以提高其他研发部门的效率，也使他们不用再重复开发组件。由此可见，这两个部门的职能是提供基础设施，提供统一的工具和平台。

营销研发部面向采销体系，采销体系主要是做网上营销的，是我们整个公司的火车头。我们为采销体系开发 3 大系统：一是交易系统，保证交易系统的稳定，同时实现网站转换率提升、用户体验优化；二是供应链系统，与供应商连接，对订货量进行预测，很多相关的供应链控制手段也是在该部门实现的；三是开放平台，要把整个供应链系统开放给第三方卖家。

运营研发部是针对 COO 体系，也就是订单生产体系的，仓储系统、配送系统、客服系统都是由运营研发部负责的。

职能研发部主要是职能体系的内部信息化管理部门，涉及诸如行政、人事、战略等职能体系，也包括我们的财务部门。

营销研发部、运营研发部以及职能研发部这 3 大部门支撑了我们内部所有的业务。而其他部门则都是为这 3 个研发部门服务的。

大数据部提供统一的大数据技术平台。数据平台是基于 Hadoop 的，在这上面做统一数据的采集、抽取、存储、处理、挖掘，以及在这上面开发一些数据增值产品。我们的搜索、推荐系统、开放数据服务都是在这个部门完成的。

移动部负责客户端的建设和移动的创新，是一个发展非常快的事业部。

研究院是一个统称，我们有一个专门的部门负责全国研究院的建设。

最后，我们还有一个部门，类似综合管理部，叫技术研发管理部，主要负责

管理体系的建设,包括 SQA、PMO、一个 IT 服务台,以及 24 小时的监控与运营。我们有一个 400 电话,系统有问题的时候,外部和内部都可以通过打这个电话进入处理流程,先由运营人员处理,处理不了的交给研发部门。之所以把研发部门往后放,是因为研发人员是不愿意直接处理问题的,他们更愿意写代码。通过运营部门的过滤,80%的系统问题都已被直接处理。

整个结构调整,我觉得还是比较成功的,大大提升了客户的满意度。以前的客户跟我们很多部门沟通,现在只需要跟一个部门沟通,而每一个部门的客户都是明确的,所以部门的主要职责可聚焦为提高客户满意度。

虽然我们有 9 个部门,但实际上我们把整个研发团队分成了 3 个层次 4 个平台:

**第一层是技术平台**,主要基于之前提到的云平台和运维这两个部门,打造基于云的技术架构,支撑其上所有的应用。

**第二层有两块**,首先是大数据平台。我们把数据独立出来,不能让每个应用都处理数据。如果每个应用都处理数据的话,这些数据就乱了,而且这些数据可能形成孤岛,很难共享,很难在整个企业里面流通。**其次是电商开放 API 平台**,我们把电商的核心平台建立起来,同时以 API 服务的方式进行服务,这对价值链长、流程和系统复杂的公司来说是很重要的。

**第三层是应用平台**。因为有了以上两层作为支撑,在上面做具体应用就非常方便了。所以我们的应用平台,包括我们自己的网站、移动客户端、内部的 ERP 及外部的 ISV 开发应用,都可以调用电商核心 API,同时相应的数据都进入大数据平台。而且,这些应用都可以在应用平台上实现。

在京东,我们针对企业架构有一些比较好的实践。大家可以对比一下,参照这样的模型打造我们的信息系统有什么好处。以前我们要开发一个需求时,不得不从前端的网站开始梳理,到采销系统,再到仓储、配送。因为整个价值链很长,需要大量的沟通和配合。现在通过电商开放 API 就很简单了,减少了各个系统之间的沟通和耦合。由于电商业务发展比较快,这种开发应用的模式更适合业务部门快速变化的需求。

通过这 4 个平台,大家看到这里有一个逻辑,最底层的实际上是我们的技术架构,用来解决架构的问题。其上的数据这块起数据治理作用,API 平台解决服务治理问题。开发都是基于 SOA 的,大家都使用某个服务的时候,就会出现混乱,需要治理平台来加以解决。这三个关键把握住,应用平台上面的应用可以丰富多彩,我们的系统逻辑也会比较清晰。

## IT 管理 123：1 个愿景、2 个重点、3 个体系

MBA、EMBA 这些理论都是来自生产线，从一线工人那里来的，这些管理理论现在对 IT 来讲都不成熟，所以在 IT 管理方面我们正在探索。

首先，要有一个愿景。几十人的团队靠身体力行，几百人的团队靠体系，几千人的团队要靠愿景。有愿景以后团队就有使命感、自豪感和成就感。

两个重点工作，一是文化。团队大了以后，文化、氛围是最重要的。

对整个公司来讲，比如老刘，再如其他业务部门的高管，他们能不能尊重技术，公司能不能形成尊重技术的氛围，这是公司的技术总负责人要考虑的。我要打造一个环境，尊重这些专家。

在研发部内部，我提了 3 个词，第一是信任，第二是分享，第三是成长。

信任就是授权，信任可以减少沟通的成本，上级对下级、下级对上级、同级之间都要形成信任的氛围。比如说研发人员有可能制造了一个事故，你认为他是故意的，还是为了创新？你违背流程，犯错当然是要惩罚的。然而京东定了一条规矩，因为创新出错，我们是宽容不惩罚的。宽容失败在研发体系里可以形成一个氛围，这也基于对员工的信任。

分享是指分享最佳实践。4000 人的团队里面一定有一些人在某方面做得很好，我会让他们分享，供其他人学习。我们也提倡容忍失败，从错误当中吸取教训，获得经验。从最佳实践学习，从错误当中学习都是为了分享经验，为了进一步提高。开发人员每天都写代码，会有一些值得骄傲的代码，去年我们开展了一个活动，让他们秀出来给其他人看。这些代码不需要是一个系统，可能只是自己的一点心得。每周五下午的这个活动，非常火爆。这样程序员在写代码的时候，就想写出好的代码，在学习别人代码的时候，就想模仿别人的长处。我们内部有一个类似 GitHub 的软件，可以让程序员把分享的代码放在上面。分享对研发人员是很重要的，可以帮助研发人员快速成长。

关于成长，因为研发人员普遍年轻，普遍希望学习和成长，我们就在内部打造成成长的文化，包括给他们培训。京东的技术人员是按 T1~T6 分级的。T1 是学校刚出来的，T2 是毕业两三年但水平还不足的，我们让 T3、T4 给他们授课，每天培训，让他们成长。我们也从外面请一些老师给我们讲课。比如，有一年做了一个 UML 的培训，效果非常好，大家学到一些架构和思维的方法，掌握了一些工具。我们又开始对研发经理进行培训，大家都踊跃参加，他们提升了，我们的研发效率就提升了，创新就增加了。



两个重点的另一个就是结构。研发管理中有很多问题是结构的问题。

团队到几千人的时候，要特别关注结构。刚才谈到人才结构，人才结构就是梯队的问题，人才梯队是非常重要的。以前低级别的人才多，T3、T4、T5 太少了。为此曾发微博招聘，收到几百份简历。关于人才结构我们还做了一件事，把管理和技术分开，以前技术的人到一定程度必须转到管理，待遇才能提高，现在我们区分开，技术人员也分级，在结构上就解决了很多的问题。还有一个结构就是我们的组织结构，前文曾提到对其的调整，改为按客户方向划分，也非常成功。

最后再谈谈 3 个体系。

第一个就是产品体系。我们公司的体验，不管是消费者的体验、第三方卖家的体验、供应商的体验，还是内部业务部门的体验，都是产品驱动的。我们打造让产品经理说了算的文化。以前产品做什么，首先是被业务部门绑架，业务部门说你这个东西要做成什么样的，产品经理记下来，然后直接找研发做，这是有问题的。另外，我们的管理者想把一个东西做好，主观上会按个人的好恶指挥产品，这样也把产品经理绑架了。然后，开发人员说这样设计太麻烦，那样做更好，又绑架一次。3 个绑架像 3 座大山，把产品经理毁灭了。我们提出让产品经理说了算，不是管理者说了算，不是业务部门说了算。比如说每个采销部门，都想定搜索排序的规则，但是我们内部有一个铁的纪律——搜索的规则只为转化率负责。业务部门告诉你的只是建议，产品经理就有自己发挥的空间，能为最终的用户体验负责，这是非常好的。把产品体系打造出来后，我们也成立了产品委员会，再选出委员会的常务委员。

第二个是架构体系。架构让架构师说了算，不是管理者说了算。管理者往往认为自己在专业方面很厉害，但实际他也有天花板。为让架构师说了算，我们成立了架构委员会，跟产品委员会一样，由委员会决定京东的架构是什么样的。京东技术架构的规划、实施和评审，都是架构委员会的工作。

第三个就是管理体系。管理体系有两方面，第一方面是项目管理，就是加大项目经理的权力，实行项目经理负责制。一旦项目立项了，项目经理的权力很大，对项目，包括进度、质量、投入产出，都是直接负责的。项目部有奖金，奖金驱动项目，项目奖金怎么分配，项目经理有决定权，并且权力很大，研发人员到项目里面，考核也由项目经理说了算。我们有一个项目管理平台，在上面所有项目的情况一目了然。而且，项目里面的每个人、每项工作，都可以按人和部门列出来，可按人、项目和部门统计，人效一清二楚。所以大的团队要靠系统进行管理。管理体系的第二个方面是领导力系统。京东 4000 人的团队，有 12 个部门，每个

部门 300 以上，300 人的管理难度是很大的。难度主要来自管人，我管的人不能超过 14 个，多了就没法管了。管人必须要有影响力，要有领导力。我们在领导力上是很薄弱的，因为很多技术管理者都是从专家转过来的，所以用的语言都是计算机语言，都是专业的语言，EQ 普遍不高，但是领导力系统对 EQ 要求很高，特别是跟业务部门沟通的时候。

“管理 123”是我的心得，也是我们研发体系贯彻的东西，效果还是不错的。

## 给团队一个愿景：做中国最有影响力的技术团队

京东的愿景原来是“让购物变得简单快乐”，今年做了一个调整，改成“让生活变得简单快乐”。而针对 4000 人的技术研发团队，在服从公司愿景前提下，我们提出了一个更符合京东研发人的愿景：“做中国最具影响力的技术团队”。这个口号提出来以后，我们的研发人员挺高兴的，感觉有一种自豪感，更有一种成就感和使命感，觉得这比较符合我们工程师的愿望。京东在研发上的投入每年都有增加，在中国企业的研发团队里面，我们算是投入比较大的。如果持续这样投入，我们研发人员有信心把技术做得更好。我们的技术人员看到自己的系统在公司发展中不断的成长，也会有很强的自豪感。

30 人的团队，管理者可能身体力行、以身作则，因为大家都看着你，你怎么干，大家跟着你，你要注意细节。300 人的时候要建一个体系，起码要有项目管理，这是很重要的，否则 300 人管起来就很困难。但是到 4000 人的时候，我觉得要有一个愿景，没有愿景这 4000 人的目标就不一样，很难团结起来。再次强调京东研发人的愿景：“做中国最具影响力的技术团队”！

# 目录

## 618

---

<b>1</b>	<b>阳光灿烂的日子——促销手记 .....</b>	<b>2</b>
	双 11 日记.....	2
	大数据平台的战斗.....	4
	交易系统的抗压会战.....	7
	JOS 备战.....	8
	青龙系统的一次意外之旅.....	9
	不平凡的 2014 年 .....	11
<b>2</b>	<b>骇客帝国——数据中心网络大优化 .....</b>	<b>12</b>
	IPO 蜜月期的 618.....	12
	核心服务 .....	13
	提升改进 .....	16
<b>3</b>	<b>超级战舰——高性能的交易系统.....</b>	<b>24</b>
	设计大流量高稳定交易系统的 10 大原则.....	24
	系统切分 .....	25

灾备切换 .....	26
灾备集群 .....	28
限流 .....	28
分流 .....	29
降级 .....	30
读写分离 .....	31
线上压力测试 .....	31
预案 .....	32
应用实例：实时价格系统.....	33

<b>4</b> 完美风暴——海量订单处理.....	36
OFC 的重要性.....	36
OFC 的形成.....	37
技术的改造 .....	38
不得不爱的运维 .....	43
海量数据的开始 .....	46
一直在路上的 OFC 人.....	50

## 产品演进

---

<b>5</b> 大时代——商城网站发展史 .....	52
青铜时代 .....	52
白银时代 .....	53
黄金时代 .....	55

不避讳差评的公司.....	58
<b>6 兄弟情——与腾讯 ECC 的融合.....</b>	<b>60</b>
写在融合前 .....	60
京东与 ECC 融合之易迅.....	62
京东与 ECC 融合之 QQ 网购.....	65
京东与 ECC 融合之拍拍.....	70
写在融合后：J+T=今天 .....	75
<b>7 猩球崛起——POP 传奇 .....</b>	<b>76</b>
POP 平台简介 .....	76
POP 平台搭建 .....	80
POP 平台演进历程.....	84
POP 新业务，托起 POP 的另一片天.....	95
京东特卖之路 .....	101
POP 生态建设新时期.....	105
POP 未来使命与荣耀并存 .....	106
<b>8 速度与激情——极致体验的供应链.....</b>	<b>108</b>
仓储系统发展历程.....	108
物流配送系统发展.....	112
<b>9 夺宝奇兵——售后系统.....</b>	<b>126</b>
追赶业务 .....	126
客户体验至上 .....	131
驱动业务 .....	133

融合开放 ..... 134

新模式 ..... 136

**10** 华尔街——财务系统演化史 ..... 137

    京东资金的秘密 ..... 137

    结算——京东和供应商不得不说的故事 ..... 140

    发票——一张纸的前世今生 ..... 146

**11** 成长的烦恼——内部信息化进程 ..... 149

    京东特色新员工配型 ..... 149

    京东企业内部门户 ..... 150

    京东业务流程平台 ..... 155

技术演进

**12** 少年派的奇幻漂流——从.Net 到 Java ..... 160

    ASP 时代的个人网站 ..... 160

    封闭开发升级.Net..... 161

    架构升级为 Java、服务化 ..... 163

**13** 通天塔——中间件 ..... 168

    服务框架 SAF ..... 168

    工作流引擎 PAF..... 174

    京东消息中间件 JQ ..... 177

<b>14</b>	<b>变形金刚——SOA 治理</b>	<b>184</b>
	服务化开端	184
	服务拆分	186
	服务治理	189
	事件驱动	190
	流程编排	192

# 创新

<b>15</b>	<b>壮志凌云——云平台</b>	<b>198</b>
	京东云背景	198
	内部京东云	199
	京东电商开放云	204
	电商云解决方案	207
	京东智能云	209
<b>16</b>	<b>穿普拉达的女王——移动创新</b>	<b>212</b>
	虚拟试镜	212
	虚拟试衣间	216
<b>17</b>	<b>少数派报告——大数据洞察未来</b>	<b>225</b>
	京东大数据平台演变历程	225
	京东大数据产品	229
	大数据技术创新	234

京东大数据创新应用篇..... 241

**18 人工智能——智慧的采销 ..... 251**

人工智能的魅力 ..... 251

智慧选品与智慧定价..... 252

供应链销量预测 ..... 255

自动补货 ..... 262

数据罗盘 ..... 270

**19 蝴蝶效应——创新孵化 ..... 275**

创新是一种价值观..... 275

“京东网站未来五年畅想” ..... 276

孵化器诞生 ..... 278

4 个“蛋”..... 280

“微创新奖” ..... 282

创新是一种文化 ..... 284

牛人

**20 神盾局——京东技术大咖..... 286**

开拓者——邓正平 ..... 286

百变神龙——杨思勇 ..... 288

代码狂——赵刚 ..... 289

青龙之父——李东 ..... 291

纵云者——刘海锋 ..... 292



灯塔——彭青 .....	294
强迫症患者——张克房 .....	295
千库之魂——樊建刚 .....	297
数据达人——刘思喆 .....	298
思考者——隋剑峰 .....	300
问题终结者——王春明 .....	302
四驾马车的车轮——陈成钢 .....	303
知心爱人——刘尚堃 .....	305
618 之盾——王晓钟 .....	307
创新先驱——杨凯 .....	308

618网上购物节的树立可以说是京东的成人礼，也给京东研发带来了每年一次的大考，更为京东技术架构提供了飞跃的机会。

618会战团队如何进行核心网络技术/订单系统/交易系统的扩展升级、又如何度过那深刻难忘的不眠之夜，本篇为您娓娓道来。

- 促销手记
- 数据中心网络大优化
- 高性能的交易系统
- 海量订单处理



# 1 阳光灿烂的日子

## ——促销手记

### 双 11 日记

在整理《京东技术解密》这本书的这一章节时，我们采访了很多位深度参与京东历年 618 和双 11 大促活动的技术人，听他们讲了很多突破想象的故事。但是，几乎所有的人，都不约而同地多次用“战场”、“战斗”来形容自己当时的工作状态，这也是我们现在回看京东过去十年的技术发展历程时，能想到的最为贴切的词汇。

2013 年 11 月 8 日，周五，距离双 11 还有 3 天时间

“这一次不能挂！”，这是周末下班前飞哥对我们说的一句话。说这句话的时候，飞哥眉头紧锁，表情凝重。一只手暗暗地握紧了拳头，另一只手坦然地接过了我递过去的橙子。“公司又开始发水果了啊，这么给力！”。

没有人可以否认，双 11 很重要。“对于网民，这是一个举国狂欢的日子；对于我们，这是一场没有硝烟的战争。这一天，在所有的人都以为我们和往常一样坐在电脑前的时候，其实我们早已化身战士，身披黄金战甲。我们的面前，是一群兵强马壮敌人；我们的背后，是支持我们的可爱用户。战争结束后，我们要

和用户一起分享丰收的喜悦……”。

“拉倒吧，貌似去年的双 11 你躺在床上睡大觉呢吧！”，正当我沉浸在美好的回忆之中，贤军同学的话把我带回现实。“哪有，去年的双 11 我明明在家里上京东下了一大笔订单……”。虽然去年双 11 没有来公司值班，但是在第二天上班期间看到每一个兄弟还在竭尽全力坚守岗位，已经废寝忘食通宵达旦，那一场景让我至今难忘。

## 2013 年 11 月 10 日，周日，距离双 11 还有 1 天时间

周末的地铁里人来人往，最显眼的当然是地铁两侧京东和天猫的广告，大胡子男人和被晒黑了的女人用各自独有的方式展示了“双 11 不能等”的心情。地铁门开了，迫不及待的人群瞬间把我冲上了地铁，在进入车厢的那一瞬间，有一种终于抢到宝贝的成就感。

晚上 8 点，维多利亚的会议室里人山人海，以老马为总指挥的战斗指挥部，每一个人都精神饱满、斗志昂扬。我们的飞哥更是一心想打破他当年连续 72 小时不下火线的纪录，当然，我们如今的系统已不是当年的系统，他已没有破纪录的机会了。

## 2013 年 11 月 11 日，周一，双 11 当天

这一天就这样到来了，早上睁眼看手机，日期处赫然写着：11 月 11 日。

比平时早一小时到公司，发现大家竟然都到了。“龙哥，赶紧观察线上日志和监控系统”，还没有来得及坐下，就已经有任务下达了。从大半个月前我们小组成员就开始精心备战，包括代码的优化、系统的调优、架构的升级、接口调用量估算、线上日记地址及监控统计……今天终于正式迎来了对我们的检阅。

11 月的北京已经进入了初冬，办公室里却涌动出热浪，在空调呼呼的背景音下，每个人都在目不转睛地盯着电脑屏幕。偶尔有人摘下眼镜，用衣服的一角擦拭一下镜片，又继续抬起头盯着屏幕。

“量上来了，又上来了！”，有存同学的大嗓门时不时地在提醒我们今天是一个特殊的日子；老耿已经在自己的办公桌和 DBA 办公室之间辗转来回多趟，终于在第 N 次回来后看见他脸上露出了舒心的笑容；老陶那边的促销是挑战最大的，各种促销秒杀活动，除了考验系统的支撑强度，还在考验着他们的心理素质；世超同学则不停地接着电话，不断解释诸如为什么库存一直同步不了等问题；而洪来同学则在忙着解决商家商品线上的各种问题……

这一刻，每一个人都十分忙碌。

## 2013 年 11 月 12 号，周二，双 11 结束的第 2 天

在双 11 结束的凌晨，看到好多同事发表了欢送双 11 的状态。日流量新高，日订单量新高，日配送量新高……各式各样的数据表明，我们再次创造了奇迹。

POP（开放平台商家）商家客户端在双 11 期间没有出现任何问题；双 11 当天，各系统中心单日接口调用量均创新高；商家中心、商品中心核心接口调用量均达到预期，且在处理关键的库存和价格时，响应稳定及时；双 11 期间，参与促销的商家占 96%，近千万款商品参与了促销，所有接口较平日调用量均有大幅度增长。各种各样的数据表明，在京东流量上涨的同时，我们的系统经受住了巨大的考验。

除此之外，这期间还涌现出了各种感人事迹。贤军同学主动请缨留在公司通宵值守；飞哥在发烧的情况下，全然不顾会传染给我们的风险，毅然坚持加班到半夜；陈川和欧阳同学在下班的路上得知系统报警后，在公司和老婆之间坚决地选择了公司……

非常喜欢一本书中的一句话，“你是一个什么样的人，你就会遇到什么样的战友！你相信梦想，梦想就会相信你！”我很庆幸我的身边有这样一群战友，我也相信这次的双 11 绝不会是一个终点，它是我们又一个新的起点。

下一个双 11，下一个 10 年……愿我们永远同在！

## 大数据平台的战斗

2013 年的双 11，对于数据部来说，他们领到的任务是为后台商家提供双 11 销售成果的关键分析数据，以及给每个商家提供云图功能的实时数据。最重要的是，必须以最快的响应速度向公司提供双 11 的最终销售数据和流量数据。在这个过程中，挑战无处不在。

## 11月10日 17点~20点

双11前夜备战，大家都摩拳擦掌，早早进入了战备状态。

## 11月10日 20点~22点

网站浏览量快速上升，已经是平时的两倍多。商家实时销售量、实时访客的前端展现服务器的资源消耗猛涨了上去，按这个趋势发展下去前端实时报表服务扛不了多长时间了。不过兄弟们在双11前都制定了紧急预案，大家立刻开始了预案实施。实时展现的兄弟们有条不紊地切换了另一套展现方案，实时系统的兄弟紧接着做了数据推送接口程序的切换。忙碌之后，实时报表展现的服务器压力降下来了，前端展现也没有出现任何时间的数据断点，也就是说，丝毫没有影响商家对自己实时销售、浏览情况的追踪。

## 11月10日 22点~24点

第一波用户流量高峰到达，从实时数据上看PV接近618和去年双11的两倍。我们抗住了第一波高峰！

## 11月11日 0点~4点

实时系统平稳运行，现在是Hadoop批处理任务上场的时间。

订单数据，对公司来说是最重要的计算销售指标的数据。过了零点，开始抽取线上数据的流程。10分钟后，数据仓库的同事发现订单表数据抽取速度过慢，还不到平时的1/5。问题来了，我们通知运维找DBA的同事和网络同事排查问题，最终在维多利亚双11指挥部，SQL Server DBA的同事发现我们抽取的数据源网络有问题，因此紧急切换了问题数据源。很快我们就顺利地抽取了线上的生产数据，Hadoop数据仓库运转正常，并在早上8点前完成了前一天所有数据的各项核心指标。

## 11月11日 8点~12点

白天的高峰再次袭来，这次的时间更长，更猛烈。实时系统平稳运行，计算着每时每刻的用户浏览量，从数据趋势来看，今天的PV毫无疑问又一次超过了行业和平均增长幅度。

## 11月11日 12点~20点

浏览量还在急速上升，用户浏览日志服务器的同事说要扛不住了，需要紧急

增加 1/3 的日志服务器。实时系统和 Hadoop 批处理的兄弟们又开始了数据接入的准备工作。18:30，日志系统准备好了新的日志服务器上线。19:00，成功测试完成，开始切换。大家盯着实时报表，时刻观察着切换可能引起的数据异常。10 分钟过去了，20 分钟过去了……浏览数据在晚高峰中平稳地增长着，切换成功了。

### 11 月 11 日 20 点~24 点

今天大家都在，为了见证京东又一个销售和流量的高峰。在大屏幕前，流量还在一点一点地上升。大家围着我们大数据平台的总监，讨论着这一年的进步。Hadoop 数据仓库从无到有；实时系统在关键时刻越来越稳健的运行；数据传输工具从第一代发展到了第二代，现在可以支撑每天数百 GB 异构数据源数据传输；调度系统、监控系统更加完善；团队也日益壮大。整个数据仓库的生态系统在茁壮地成长着，未来我们京东的大数据蓝图也在今天实现了很大程度的预演。临近 24 点，我们在实时报表前合影留念，以纪念这个难忘的双 11。

### 11 月 12 日 0 点~3 点

现在最紧张的时间到了，昨天到底有多少销售量，有多少 PV 的谜团要揭开了。

仅用 80 分钟大家就把数据量已经超过 1 个亿订单表的数据从生产 SQL Server 抽到了 Hadoop 上。

又过了 30 分钟，浏览日志数据在 Hadoop 上处理完成。

### 11 月 12 日 3 点~6 点

Hadoop 数据仓库开始了计算流程，核心指标以最快的速度计算完成。在各个数据需求方 9 点钟上班的时候，已经能够看到了京东双 11 的各项核心指标了。

此刻，新一天的太阳升起。晨曦中，数据仓库的同事终于可以回家安心地睡觉了。在大家休息的时候，我们在战斗；在大家看到胜利成果的时候，我们已经离去。这就是京东数据部大数据平台同事的双 11，也是每个研发同事的双 11！

## 交易系统的抗压会战

每次大促过后，交易系统的上游和下游在高压情况下，都会暴露出一些不同的问题。目前一年两次的大促活动，老宋所在的交易组的伙伴们，考虑的最多的就是，如何避免 618 遇到的问题在双 11 重演，如何应付双 11 出现的新状况，如何提前做预案等问题。

老宋回想着今年的 618，交易组系统的上游系统遇到瓶颈，购物车在不停地关闭一台服务器，再打开另一台服务器，然后重启上一台关闭的服务器，一直这样，不停地重启、再重启。否则购物车服务器马上就会遇到瓶颈，导致用户进入购物车后无法正常下单。双 11 前夕，交易组领导老王签下军令状，首要任务就是优化交易系统性能。为此，老王亲自组建了虚拟性能小组，开始调整 Nginx 配置、合并异步调用减轻购物车压力、优化用户和商品接口、实现内存计算防刷以及进行各种降级方案、备节点、异步节点和亦庄机房的多次演练等，将很多性能方面的工作都当成了日常任务解决。

双 11 接受检验前夕，在加压测试中发现，经过优化的上游系统表现出色，购物车和下单系统没有遇到任何瓶颈，之前性能优化的工作已经产生了效果，淋漓尽致地发挥了作用。但是，下游系统又遇到了较大的挑战，产生了比如订单任务积压、很多支付用户订单状态不能流转、3COD 挂掉导致 SOP 订单不能选择货到付款等问题。如果真的到了双 11，这些问题可能又将导致一次次糟糕的商家和用户体验。好在经过提前的预演，老王可以胸有成竹地开始第二轮带着团队不断解决系统在高压下存在的各种问题。

下游系统出现的问题与系统架构关系密切，每次老王开始讲解订单任务和中间件的系统架构时，都会画出很庞大的交错复杂的架构图。老王时常在想，我们一定要做减法，给下游系统瘦身；干掉冗余的部分，让一些逻辑关系迁移出去……架构调整的压力已兵临城下。



## JOS 备战

进入6月，整个进出北辰大厦的人，好像走路的节奏都变快了，虽然整座大厦的冷气让室内温度降到了22度，但是你会感到618似乎带着京城40度的高温向你一股一股地扑面而来。

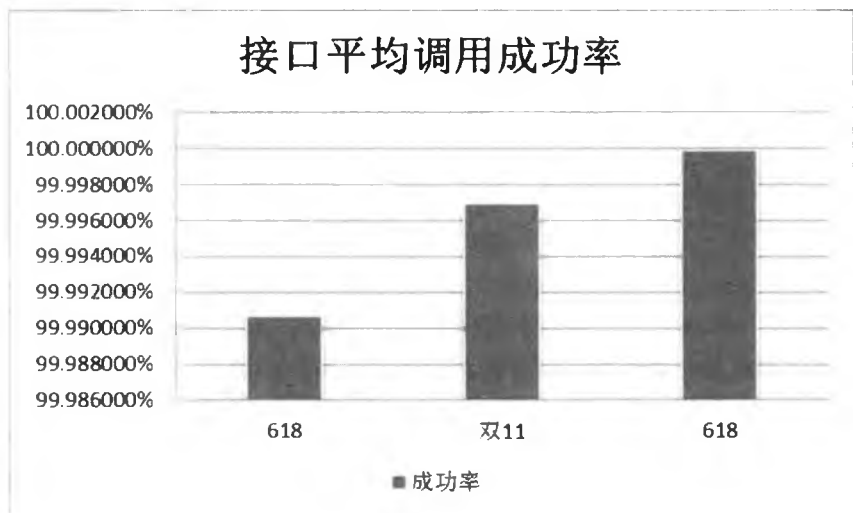
JOS 开放平台备战。JOS 开放平台是京东对外的一个门户，也是众多后端服务接口正常调用的重要桥梁之一，JOS 的团队清醒地认识到了这一点。JOS 平台负责 API 开放的老黄认为，JOS 系统的负载时刻与 618 活动脉搏相通相连，用户的每一次授权登录对我们来说，都是最好回馈；商家的每一次活动上架都实时挑动着团队成员的每根神经；商家最后的欢喜出库也在考验着 JOS 团队这几个月来不懈的付出。为了今年的 618 大促，JOS 团队已经提前几个月开始准备了。

经历了 2013 年的 618 和双 11，JOS 网关在整体稳定性上已经有了明显提升。但是为了迎接 2014 年的 618，JOS 团队仍然对网关的效率和性能做了整体分析和评估，他们认为 JOS 系统需要做综合的升级处理。老黄认为，我们首先应考虑的是稳，618 的第一宗旨是“稳定压倒一切”。但是现在，对 JOS 系统的后端服务接口评估后发现又快有慢，这一点必须调整、完善。

不仅 JOS 开放平台后端服务的一部分接口处理时间有快有慢，而且一些系统的容量规划、发布时间和服务质量也有参差不齐的情况。考虑到开放平台作为一个对外的门户，HTTP 请求的同步性+容器管理线程生命周期，将会使得单个服务不可用波及平台，最终导致后端服务都无法被访问，因此，这里需要改变传统容器的请求处理方式。（假如一个服务出现问题，同时 3 秒钟为服务调用超时时间，如果一个应用服务器最大容器线程为 600 个，那么单机最差情况 1 秒就只能处理  $600/3$  个请求，这意味正常的后端服务由于得不到路由中转，也被外部认为不可用）将容器线程池独立出来处理连接，而业务处理交给后端业务线程池处理，业务线程池可以根据业务优先级设定一些预留和限制模型，即共享线程池，使限制线程池被独占。JOS 团队做了“异步模型封装+业务管道化+业务权重线程池”的处理，异步模型设计提升了系统的响应时间，而配合权重线程池处理则实现服务之间质量影响隔离以及服务与平台稳定性隔离，通过隔离保证了系统的稳定运行。

在今年 618 的总结中，JOS 平台负责的 618 前后 3 天的接口调用总量与去年

618 和双 11 的调用总量相比，高出了近 3 倍。而接口平均响应速度比去年 618 提升 82%，比去年双 11 提升了 25.6%。



2014 年 618（右）与 2013 年 618（左）和双 11（中）的接口平均调用成功率比较

## 青龙系统的一次意外之旅

2014 年 6 月 19 日 16 层成都会议室

整个指挥中心，除了通宵后仍然坚守在监控岗位的值班同事，最抢眼的就是墙壁和桌面上各个生产系统的监控图表了——流量监控、服务器性能监控、接口性能监控……各种监控，有图有真相，一旦哪里突发异常，系统负责人立刻会做出响应。对于青龙系统，2014 年 6 月 19 日是系统负载最大的一天，订单的洪峰通常晚一天到达配送，从早上 5 点后，系统访问量开始直线上涨，这段时间大家重点监控数据库的负载、Worker 任务的积压情况。直到早上 8 点后访问量开始下降，系统性能指标一切正常，大家悬着的一颗心终于放下了。为了今年的 618 大促，我们提前几个月就开始了准备，回想整个过程，有意外也有惊喜，但更多的

是充满了荷尔蒙的刺激。

经历了 2013 年的 618 和双 11，青龙系统稳定的度过了一段比较长的时间。为了备战今年的 618，我们决定将数据库安全等级提高，2014 年 4 月 5 日凌晨开始修改数据库密码，青龙上下一共几十个应用，按照系统的依赖顺序编写了一份操作步骤。按照这个步骤，凌晨的切换非常顺利，直到早晨业务高峰前系统都运行正常。当大家认为系统已经运行稳定的时候，忽然收到报警，紧接着大家就一直埋头处理报警，直到下午才完全解决问题。发现是切换的操作步骤出现了问题，导致各系统间依赖访问量比平时大了很多，访问异常会再次出现恶性循环的重试，直到超过了系统最大的负载。

发现这个问题后，我们反思，如果我们的系统自我保护能力很强，或者我们的系统有降级方案，又或者我们有完备的应急预案……那么这个问题就不会这么轻易地危及到系统的正常运行，IPO 在即，618 在即，警钟已经敲响，我们必须行动。用一句话描述我们当时的状况，那就是“自助者天助之”。基于这套我们自己开发的系统，也是我们最为清楚和熟悉的系统，我们对青龙运单展开了一系列的可能出问题点的设想。

1. MemCache 缓存单点的问题，一旦性能出现问题，扩展不方便，且现在没有专门的维护团队；
2. 依赖底层基本资料服务问题，假如基本资料不可用，运单本身的部分服务也间接受到影响；
3. RAC 生产库一旦出现宕机或性能问题，本身的性能会受到很大影响，间接影响到其他上游业务系统。

针对这些风险点，我们组内进行了多轮的头脑风暴，最终得出一套切实可行的解决方案。

经过了短时间的开发，第一阶段的成果在 2014 年 5 月初顺利完成上线。紧接着，我们开始了第二阶段的开发改造。第二阶段很重要，不仅留给我们的时间紧，而且修改点覆盖面广。从 2014 年 5 月中旬开始，对新的一版程序进行线上发布。这次发布后，我们专门对数据库的性能做了分析，发现加了全量缓存后，数据库的负载不但没有降低，反而升高了。经过分析最后认定是业务流水表缓存刷新的问题导致，最后针对业务量大、访问量少的数据缓存机制做了策略调整。再次发布后，数据库负载明显降低，缓存命中率直线上升。系统优化的同时，我们还编写了切实可行的系统应急预案，配套本次的优化改进。在所有优化点都上线后，经过评估，我们认定即使单量翻几番，系统也可以平稳的应对了。

## 不平凡的 2014 年

“2014 年对于京东来说，注定是不平凡的一年”，李大学在今年的 618 会战中说道：“京东的技术研发团队在每一年的 618 实战中，都能够得到充分的历练。可以看到，我们的系统一年比一年完善，战绩一年比一年精彩。能参与 618 战斗的技术人是幸运的，如果要问我一个计算机工程师该以怎样的态度和方式来工作和学习？我有一条朴素的建议，也许能够很好地解答，那就是——参与 618，发现问题，解决问题。不要绕开问题本身，多做事情，不会吃亏，即使不是你的事情。”这看似简单的建议，其实并不是很容易能够做到。但是，2014 年的 618，研发体系真正做到了这一点。（作者/马沛、许立龙、张侃、李刚、王先科）

# 2

## 骇客帝国——数据中心

### 网络大优化

#### IPO 蜜月期的 618

2014 年的 618，对于京东来说是一个具有非同寻常意义的 618。

2014 年 5 月 22 日，京东正式在美国纳斯达克挂牌上市。本来 618 对于京东研发来说，是每年一次的自我检验、持续提升的过程。但今年 618 却成为了京东 IPO 后需要向市场和投资者交付的第一张重要答卷，而京东研发所在的技术体系则是完成这项工作的重要环节。

仍处于 IPO 蜜月期的京东面临着一系列的挑战。首先，是 IPO 之后来自社会各方的压力。与往年不同，这次京东确实需要通过一个完美的 618 盛典来巩固自己的行业地位和增强投资者的信心。其次，是与易迅、微信等业务的融合。京东在 IPO 之前与腾讯迅速达成了战略合作关系，京东需要快速将战略合作成果落地，将资源和人气转化为收益。最后，是自身业务高速发展带来的压力。京东销售额连续多年以高于行业平均增速的速度增长，其影响直接体现在网站访问量、成交量和网络流量的倍数增长上。京东研发需要在满足业务高速增长的情况下，完成对架构和系统的快速升级和持续优化。

挑战和压力是客观存在的，虽然并不会因 IPO 而发生本质上的改变，但京东公司上上下下都意识到了这次 618 的特殊性和重要性，以及公司内外对这次 618 的殷切期望，因此，京东研发系统在备战 618 过程中，给予了更多的重视和投入。

618 每年只有一次，但是从 2013 年 618 刚刚结束之际，京东研发的 2014 年 618 备战工作就已经开始了。

## 核心服务

笔者所在的部门是隶属于京东研发中心运维部的基础架构部，主要负责面向数据中心的系统和网络基础平台的运营工作。

其中，京东数据中心网络，包含 3 个核心服务：CDW、负载均衡、基础网络。这 3 个服务几乎承载了京东所有的业务和流量，属于最关键的基础网络服务，直接关系到业务系统的服务能力和质量。也正因为他们属于支撑其他业务系统的基础平台，导致在建设周期、方案优化和架构升级等方面，受到更多的制约和限制。在业务爆炸式增长时期，基础架构更多满足的是业务高速增长需求，为此在架构可用性、扩展性等方面留下了很多问题和遗憾，进而影响了基础架构服务的长期可持续发展。随着业务系统的进一步增长、数据中心规模的加速膨胀，这些遗留问题的负面影响就愈发地突显出来，甚至造成业务应用服务质量和可用性的明显下降。

除了历史原因留下的旧伤，还有互联网技术发展带来的新需求。前端应用的技术演进、分布式系统的大量应用、云计算和 SDN 等新技术的推广，对基础架构提出了前所未有的挑战。这三项网络服务，都面对着看似各不相同的问题。



CDN、负载均衡、基础网络面对的主要问题

## CDN 面对的问题

2009年，运维部元老 Gavin 在对 CDN 几乎零经验的情况下，通过在网上查找资料自学、大量实验和环境测试，搭建了基于 Squid 实现的第一版京东 CDN 服务。出于好奇我特别找到 Gavin 了解当时决定自建 CDN 的原因，原来是因为当时使用的商用 CDN 服务出现了几次严重的故障，促使 Gavin 决定自己来做，看来自己动手的基因真的是深入互联网公司的骨髓。

在无法获得更多外部支持的情况下，CDN 团队几位成员自学自建，在不断摸索的过程中逐步发展和完善着 CDN 服务，京东 CDN 的规模随着业务和流量的快速增长也越发突显出其重要性。2013年上半年，京东 CDN 完成了全部节点从 Squid 向 ATS 的迁移，全国总共几十个 CDN 节点，承载了 90% 的互联网流量，当年 618 互联网峰值流量超过 100G。

相信很多企业都有过类似从无到有的成长经历，无论是对企业还是个人，这都是极其宝贵的财富。在信息高度发达的今天，我们可以通过各种渠道获得丰富的理论知识、技术方案和应用案例，但在面临诸多需求和选择的情况下，能够准确地把握核心需求、找到适合自身的最佳选择、确立中长期发展的方向和策略，恰恰需要在专业领域长期的实践和积累。

但是，从 2013 年开始，CDN 产品的一些问题也逐渐暴露出来，集群性能不足、系统结构缺陷、扩展性较差等等，这些问题仅仅依靠升级硬件、更改软件、增加带宽已经远远不能解决了。另外，早期出于成本、需求等多方面考虑，CDN 节点主要使用的是二级运营商的数据中心和网络，尽管其机房环境和服务水平远远优于一级运营商，但由于中国电信行业的特殊情况，导致二级运营商网络质量相对较差，这对京东网站的服务质量造成了致命的影响。

## 负载均衡面对的问题

京东自从开始使用负载均衡服务以来，就一直在使用商用设备。在业务上有客观需要，企业具有一定购买能力，但自身又没有负载均衡产品的独立开发能力的情况下，使用商业负载均衡产品似乎是一个自然而然的选择。

在京东的业务场景中，基本上所有面向公网的业务均通过负载均衡提供服务，与此同时面向数据中心内部的业务也大量使用了负载均衡技术，有一些确实是负载的需要，还有一些是为了容错，甚至一些 NAT 之类的网络服务也部署在负载均衡设备上。随着业务和网络规模的持续增长，负载均衡的压力也越来越大。随着时间的推移，网络中出现了不同品牌、不同型号、具有相当规模的商用负载均衡设备。

终于，负载均衡服务的各种问题开始大量出现，主要是性能和结构的问题。无论是老款设备还是新型设备，都开始出现单机性能不足的情况，为了缓解设备压力只好不断增加设备数量、选择更高性能的型号。由于商用负载均衡产品基本都是两台一组的 Active-Standby 结构，因此每增加一组设备，就要将原有设备上的业务做一次拆分，这是一个非常痛苦、不优雅的过程，其管理维护成本高、风险大、缺少弹性。

实际上，此时已经有很多互联网同行在使用 LVS 等开源负载均衡软件，并形成了许多相对成熟的解决方案，为了与商用产品区别我们称之为软件负载均衡。但是，软件负载均衡需要具有一定的开发和调优能力，否则其服务能力还无法达到替代商用产品的水平。京东一直在软件负载均衡方面积蓄力量，只是还未遇到一个适时的引爆点。

## 基础网络面对的问题

基础网络主要具有 3 个核心功能，或者称为服务。

第一项是服务器等主机设备的网络接入服务，这部分功能主要影响的是主机设备网络的有效接入带宽和可用性。虽然从 2010 年之后数据中心以太网的物理接入带宽基本都达到了 1000Mbps，但实际上由于 TOR、汇聚设备等上行链路的带宽收敛，导致主机设备的平均有效带宽远远低于 1000M。随着近些年主机设备流量的快速增长，接入带宽不足的问题愈发突显。

第二项是数据中心的互联网接入服务，这部分功能主要影响的是数据中心整体可以提供互联网服务的容量和质量。随着大型集群、分布式应用、大数据应用



等技术架构的广泛使用，尽管互联网流量在保持较高速度的持续增长，但数据中心内部交换的东西流量增长还是远远地超过了互联网的南北向流量增长，二者对比由原来的南北向流量占大多数，转变为了当前的东西向流量占大多数。二者流量的总体增长给主机接入、互联网接入同时带来极大的挑战。

第三项是数据中心之间的互联服务，这部分功能主要影响的是网络对跨数据中心的数据调度和分布式应用的支撑能力，以及相对于互联网的网络质量优化能力。如果一定要套用目前最流行和高大上的概念，那么数据中心云就是这项服务的最佳诠释。数据中心界限的模糊化、网络服务的平台化，是上层应用和网络自身发展的综合需要。而高带宽专线、基于 DWDM 的传输系统是实现目标的必经之路，多数据中心的网络架构则是决定服务性能、扩展性的核心保障。

京东在基础网络上早期投入较少，在业务快速增长时，首先需要保证的是满足设备规模的增长，在网络容量、可靠性和扩展性方面没有来得及深思熟虑，而基础架构位于 IT 服务体系底层，建设周期长、迭代更新慢，很快在资源和性能方面出现了问题。例如，部分老网络存在 10:1 收敛比的接入网络，在 2010 年后的网络设备替换时仍使用了 HSRP+STP 的结构，导致网络带宽和资源利用率变相降低，随着设备老化和规模增加，频繁出现带宽拥塞和局部故障。二级运营商的网络稳定性方面影响的不只是 CDN 业务，还影响着所有数据中心的公网业务。对于数据中心的内部网络结构，由于早期设计的缺陷，也降低了网络服务的稳定性和升级改造的可靠性，大量核心业务应用对网络的过度依赖，导致基础网络成为一个日益恶化的肿瘤，进退两难。

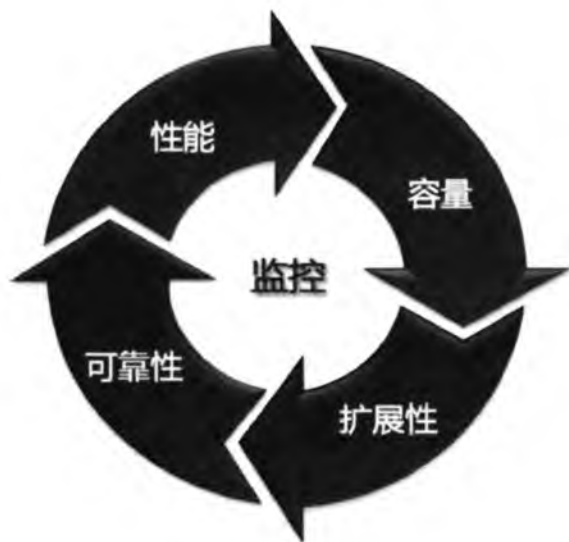
在经历了 2012 年 618 之后，我们逐步意识到，CDN、负载均衡和基础网络在上述各个方面存在问题的解决，已经迫在眉睫。

## 提升改进

### 核心问题鉴定和发展目标确立

我们经过慎重的思考和总结，回避了大量应急救火的事务干扰，将面对的各

类问题抽象为 5 大核心问题，即性能、容量、扩展性、可靠性和监控。



基础网络服务持续发展的 5 个核心问题

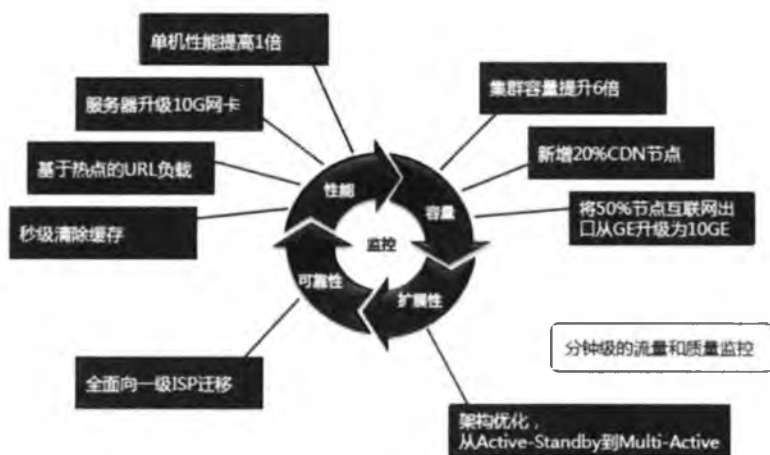
这 5 个问题，相互影响、彼此呼应，性能可以影响容量，容量规划依赖良好的扩展性，而无论多好的设计和架构如果没有足够的可靠性保障，就失去了最基本的价值和意义。而这一切，都需要通过监控系统进行分析和确认，在真实数据的基础上，掌握服务的运行状态，并根据数据和结果反向修正架构和策略，实际上这是一个服务可视化的过程。

基于以上认识，在 2013 年 618 至 2014 年 618 这一年时间里，京东在数据中心基础网络服务方面实施了一系列颇具成效的举措。

## CDN 提升

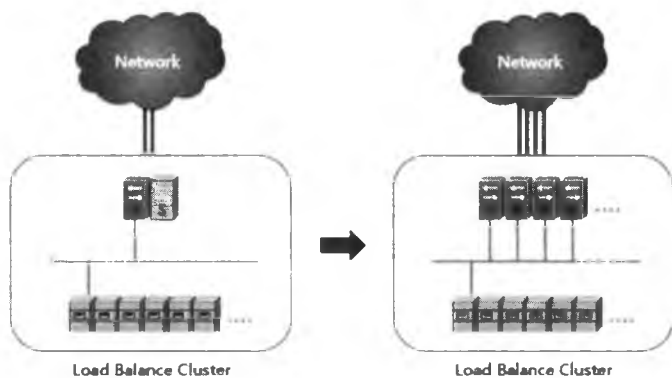
2013 年上半年京东 CDN 全部改造为 HAProxy + ATS 的结构。

接下来，首先实现了将 HAProxy 的单机性能提升一倍，在此基础上再将 HAProxy 的多个 1000M 网卡绑定方式升级为单个 10G 网卡接入，这样在除了网卡之外硬件配置不变的情况下就显著地提升了单台 HAProxy 的服务能力，这是商用产品很难达到的。



CDN 服务为支撑 618 完成的改进工作

其次，对软件负载均衡的架构进行升级，从 Active-Standby 双机主备模式转换为 Multi-Active 多主模式，这个转换是通过在集群中增加一台三层交换机来实现的。交换机完成基于流转发的 ECMP 工作，这样负载集群的容量上限就直接产生了至少几倍的增长，并且可以通过三层交换机 16-Way ECMP、32-Way ECMP 等不断提升集群的容量。



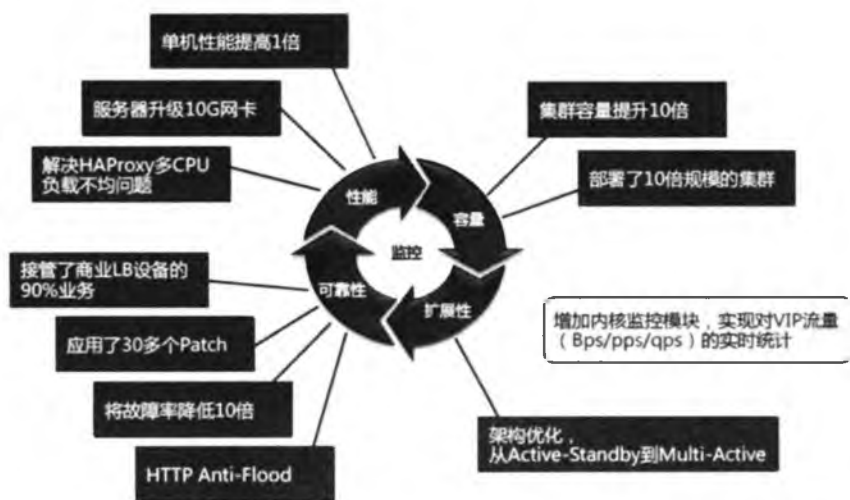
京东 LVS 集群的架构升级

到这里就不得不提到实现了上述本质改变的核心人物思兴，正是思兴加入京东之后主导了对 Linux 内核、HAProxy 和 LVS 相关产品的深度开发和优化，京东长期的积累和需求，与思兴的将想法实现的能力，像化学反应一样产生了神奇的效应。个人认为是思兴的工作成果，促使运维体系一些知觉灵敏者意识到开发工作的重要性，并将“开发”一词从纯面向前台业务突破到面向运维、面向一切需

要开发的领域，这是另外一个话题，就此打住。

在上述两项核心优化工作完成的基础上，在过去的一年里京东 CDN 单节点实际部署的能力提升了 6 倍，全网 CDN 节点数增加了 20%，并对全网超过 50% 的 CDN 进行互联网接入千兆改万兆的升级和架构升级，以及增加一级运营商网络 CDN 节点的比例。同时，针对业务需求实现了一些 CDN 功能升级，例如秒级清除缓存、热点 URL 负载等。最后，实现了对 CDN 节点流量和质量的分分钟级监控，并实现基于日志的数据分析。

## 负载均衡提升



负载均衡服务为支撑 618 完成的改进工作

京东一直在使用各类软件负载均衡产品，其中最常见的是 LVS、HAProxy 和 Nginx。实际上思兴来到京东后的第一件也是最重要一件工作，是将软负载的集群确定为 LVS + HAProxy 结构，我和 Apollo 共同参与其中。经过几次结构优化，大量的问题处理和开发，最终稳定为 Flow-based ECMP + LVS + HAProxy 的结构，这部分优化内容与 CDN 的软件负载均衡基本相同。

与商用负载均衡产品相比较，软负载集群具备以下优势：

1. 性能和容量实现几倍到几十倍的增长；
2. 具有良好的扩展性，可以按需平滑水平扩容；
3. 具有良好的可靠性，局部故障不中断服务；
4. 可快速实现新增业务和管理需求，可维护性和可视化程度显著提高；

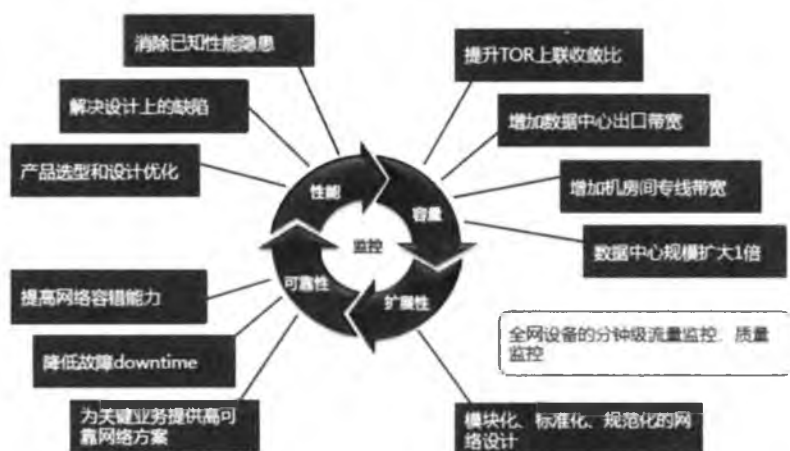
### 5. 同等性能成本显著降低, 持续服务能力显著提升。

当然, 设备数量增加、功耗增加、结构复杂度增加、管理维护成本增加也是显而易见的, 但相对收益来讲这些代价几乎可以忽略不计。

在过去的一年里, 软件负载均衡单机性能提升一倍, 集群容量提高 10 倍, 替代了 90% 的商用负载均衡设备, 将接近 90% 的业务逐步迁移到软负载产品上, 软负载集群规模扩大了 10 倍左右, 服务中断时间大幅度下降。在开发和使用过程中, 针对消除缺陷、性能提升和功能提升累计开发了三十多个补丁, 增加了内核监控模块实现对 VIP 的 Bps/pps/qps 流量监控和面向业务颗粒的监控, 解决了 HAProxy 多 CPU 负载不均、Keepalived 特定版本因 Gratuitous-ARP 实现缺陷导致 VIP 主备切换失效等相对复杂的问题, 并在 2014 年 618 之前实现了 HTTP Anti-flood 的一期功能。

思兴和其所在的系统开发团队, 是实现上述工作的关键因素, 个人认为这就是京东基础网络应用发生质变的引爆点。但不得不说明的是, 如果优质产品是种子, 那么良好运维就是种子赖以生存的土壤, 优良的产品和设计是为了满足需求和解决问题, 但脱离应用场景、偏离真实需求的产品设计容易出现闭门造车、纸上谈兵的问题。另外, 再好的产品也需要优质的运维才能保证其持续、良好的发展, 基础架构类服务更加看重产品的长久生命力和稳定性, 只有开放、包容的技术环境才能产生优质的产品和服务。

## 基础网络提升



基础网络服务为支撑 618 完成的改进工作

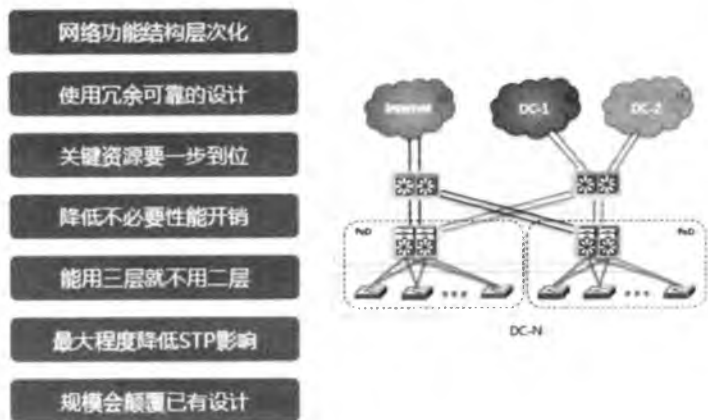
京东的数据中心基础网络，是所有已知问题中最突出的，因为它影响着所有上层服务和应用。在相当长的一段时间内，很多应用都认为带宽拥塞是导致其服务质量下降和引发故障的唯一原因。事实上一些早期网络环境中，10:1 的接入网络收敛比也确实授人以柄，而更加麻烦的是由于网络监控远远落后于网络规模的发展，导致监控的覆盖范围和准确性不足以反映网络服务的质量和运行状态，这种功能缺失造成了业务与网络相互推诿、纠缠不清的局面。另外，早期网络结构的性能不足、不可靠导致一些本来微不足道的故障和操作，被放大成影响服务的故事和复杂操作，这些都给网络优化造成一定的障碍。

为了解决以上问题，京东网络团队从两个方向着手解决问题。

首先，解决影响用户的迫切问题。

在遇到用户反馈的问题时，深入分析找到本质原因，进而确认基础网络最紧急和优先的问题。同时，自主设计和开发基于 SNMP 的网络监控工具，实现网络设备分钟级的流量监控，覆盖所有核心、汇聚、三层设备、所有接入交换机的上联接口，再覆盖一些收敛比较大的早期网络接入交换机下联端口。这样用户发生问题时，就可以准确定位网络在业务问题发生时的影响程度了。通过一段时间的积累，一方面网络团队在专业性上赢得了用户的认可和信任，另一方面也定位到一些真实需要优先解决的网络问题。于是在 2013 年双 11 前后，优先对带宽拥塞最严重的设备进行了带宽扩容，缓解了问题的影响，为架构优化改造争取了时间。

其次，从网络架构设计上根治问题。



网络运维经验总结、优化后的网络结构

最重要的改变是网络功能结构的层次化，将网络功能划分为 3 个部分：第一

是满足服务器接入需求的由核心和接入交换机构成的以 PoD 为单位的接入交换网络；第二是满足数据中心设备访问互联网的公网边界网络；第三是满足数据中心之间互联的基于 DWDM 传输系统的私网边界网络。这种功能上的分离和结构上的调整，直接收益是缩小了网络故障域，使得各个功能网络各司其职、互不影响，并且实现了网络架构设计的规范化、标准化，也实现了良好的扩展性和可靠性。

其次是对网络资源和功能配置的优化，在成本允许的范围内尽可能地实现冗余和一步到位。到 2014 年 618 前夕，京东所有新建数据中心和新改造网络，均实现接入网络 1:1 收敛比，功能网络分离，关键配件和设备至少 1+1 冗余。另外，互联网出口带宽全面从 1000M 升级至 10G，点对点专线带宽从 10G 规模升级至 100G 规模，互联网全面从二级运营商向一级运营商升级，网络整体的性能、容量、扩展性、可靠性均有了本质的提高。

## 网络服务改进

在持续服务和优化的过程中，网络团队积累了丰富的收获和体会，并形成了适合自身业务的方法论和原则。

无论有多么充分、正当的理由，都不能以牺牲用户的利益和需求为代价，服务的本质是满足用户和业务需求。有时用户需求看似与网络服务的长期规划和发展冲突，但实际上需求与发展是可以统一的。必须认清需求背后的实质问题，避免治标不治本的行为，要将用户的表面需求转换为本质需求，并通过理解、把握用户的需求指导网络服务的发展方向，同时也需要对用户需求进行理性的引导。网络服务属于 IT 服务底层的基础架构，建设周期长、影响大、牵一动百，必须有科学和长远的规划和设计才能避免不断地被短期困难影响。最后，架构设计不能回避困难、问题，必须找到本质原因、彻底解决，纵容缺陷、回避问题的架构，会造成持续的不良影响和灾难。

优先解决用户的迫切需求

平衡短期需求和长期目标

架构设计不能向问题妥协

虽然今年京东的 618 订单数与销售金额又创历史新高，但经过这么多核心服务优化，我们今年的 618 研发指挥室从容自如，系统平稳支撑。现在我们已经又开始了新一轮的规划，要为即将到来的年底大促又开始继续前进了。（作者/王大泳）



# 3

## 超级战舰——高性能的交易系统

### 设计大流量高稳定交易系统的 10 大原则

整个交易团队与几千年前老祖宗大禹面临着相同的问题，线上不稳定的高流量和洪水相似，而支持业务的众多子系统则相当于众多的村庄和堤坝。经过几年的 618、双 11 备战，团队总结出了 10 大原则，成功地抵御住了一次次流量冲击。

**第一个原则：按业务切分系统。**以京东为例，切分成首页、频道页、购物车、交易、结算，不同域名单独部署。

**第二个原则：灾备切换。**对于线上交易系统来说，平时对稳定性的要求特别高，如果你想做到高稳定性，千万不要把你的鸡蛋放在同一个篮子里。

**第三个原则：水平扩展分流。**对于线上交易系统，平时用户访问量是一个正常的量，每到 618 大促活动还有秒杀活动开始的时候，整个系统的流量就变成了平时的很多倍，几十甚至上百倍，关键是峰值来的时候怎么解决？最基本的方法就是分流。

**第四个原则：超预期流量限流。**对于线上大流量系统来说，上策是分流；中

策是限流，实在搞不定了，我们就把认为是恶意的流量挡掉；下策就是保护系统。有时候流量实在太大了，仅正常用户的流量系统可能就扛不住了，我们就挑出来一些业务点把系统保护住，保证大部分用户的体验，让其能够正常下单和比较愉快地购物。

第五个原则：降级保护系统。

第六个原则：读写分离提高性能。

第七个原则：异步化。

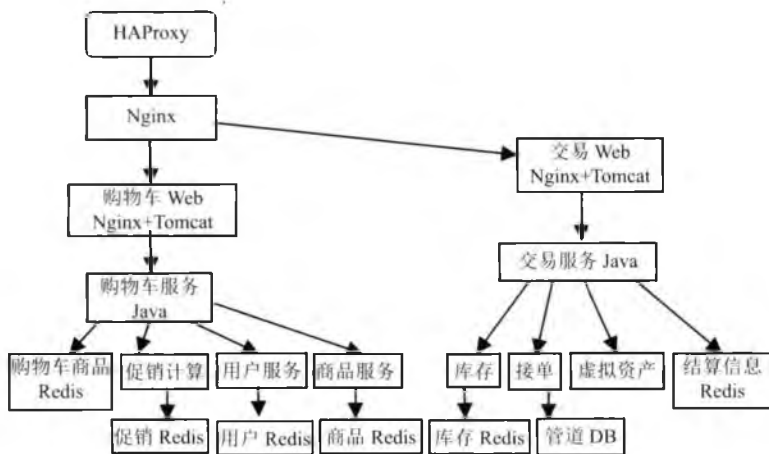
第八个原则：动态静态流量分离。

第九个原则：线上压力测试发现瓶颈。

第十个原则：预案。其实每次出故障不可怕，最可怕的是出了事你束手无策，所以，什么事都要想到前面。

接下来，我来给大家详细讲一下这 10 个原则。最后我还会拿一个实际案例来说明这 10 个原则在实际场景中的灵活应用。

## 系统切分



京东交易系统的系统架构图

首先我们按业务逻辑切分系统，如下。

**网站系统：**首页、列表、频道、单品、搜索

**交易系统：**价格、购物车、结算、支付、订单中心

**基础服务：**促销、库存、商品、用户

**订单系统：**管道、OFW、订单中间件、生产系统

这样，数据与流量都首先被分流了。

## 灾备切换

灾备切换分为两大场景，第一是多机房部署场景下的灾备切换，另外一个是一个机房内部场景下的灾备切换。

### 多机房场景下的灾备切换

京东网上交易基本上部署在3个机房，最主要机房有两个，机房一和机房二。多机房部署会面临的两个问题是机房内部出现大面积故障和机房入口流量出现异常。解决这两个问题就需要进行机房之间的自动切换，现在我们切机房的速度挺快的，基本上5分钟之内可以完成一轮机房切换。

多机房部署情况下的灾备切换有两种策略，一种是入口流量切换策略，一种是直接路由切换策略。

**入口流量切换策略：**入口流量是运营来切，通过切换把流量解析到另外一个机房去。它的好处是有地域性，可以单切东三省或者先切比较重要的地区。但是这个方式比较慢，切的系统也比较多，比如，交易系统中购物车后面有大量东西需要切换。

**直接路由切换策略：**主要应对机房出现大面积故障。其实一般机房大面积故障很少发生，这种情况下用AHbroad去切特别快，可以说是秒级的，只须从这个

脚本直接把新上的流量切走。

实际上线上出现问题，每次都是两种策略一起用，也不判断是什么原因，反正感觉有大规模故障了就双管齐下。

### 机房内部场景下切换

主要就是应用一层一层通过调用客户端来进行切换。

Java 应用层切换，我们用了自己研发的底层框架，支持调用者选择服务者类型，这种切换非常快。还有一种就是内部域名切换，切换起来比较慢。

缓存切换，京东用了大量的 Redis，我们对 Redis 客户端做了改造，可以很快地在 Redis 客户端各个级别之间相互切。数据库的切换也可以自己选客户端，有些系统客户端自己切，有一些系统可以利用数据库自身的切换机制。

路由切换起来不麻烦，真正麻烦的是怎么保证数据的一致性。比如说库存管理这样的关键业务，如出现数据不一致问题就很致命。以 iPhone 手机为例，假设降到每部 10 块钱肯定会被疯抢，这个时候如果要切换机房，期间库存到底减少多少件很难得知，极易出现卖超或者卖不动的情况。

数据之间一致性的保证有这么几种方法，第一种是程序自动切换。好处是比较灵活，商务切换特别快，机房一切换到机房二，一分钟之内肯定可以生效。而且千兆万兆不需要来回导数据，因为两边数据是在双写的。我们采用的是异步双写，异步双写数据一致性可能差一些，但性能高，线上购物车我们实际需投入大概 6 到 7 个集群，如果都是同步双写肯定是受不了。

数据一致性保证的第二种方法是底层存储数据的复制。好处是数据一致性高。比如说，Redis 制造了一套很复杂的系统，Microsoft、Oracle 的数据库也都有自己成熟的数据复制链路。其缺点是主动切换机器不灵活，比如说主节点到从节点，从节点到从节点，过程中一旦出现问题，特别难做到数据一致。比如说库存系统，必须得用数据复制，因为其对数据一致性要求较高。

## 灾备集群

先讲一下**空闲灾备集群策略**。我们有备用机房，申请的很多机器放在那里，平时上面没有任何流量。但是不能把整个交易系统都放在上面，因为会面临两个问题，一个是管理成本，一个是热量缓存问题。

业务上线后的集权管理特别麻烦。因为机器多，京东有一个非常硬性的规定，就是交易上线之后每台机器都要进行检查和备份。为避免出现网络等各方面问题，必须要人工切换，导致上线管理成本非常高。此外，空闲灾备集群还会产生热量缓存的问题。交易系统大量使用缓存，如果线上热量缓存切到一个没有跑流量的系统，每次一切换过去都会产生很大的瞬间流量。这一问题也是必须要考虑的。

第二种策略是做一个**小流量的灾备集群**。比如说在机房二的灾备集群上面适当分配较小的流量，东三省流量比较低，我们可以把东三省的流量切到机房二去。但是这也不是全无问题的，主要是会增加事故风险，而且管理成本也比较高，要跨机房上线。

最后说一下**集群切换策略**。对交易集群来说，切换策略可任选，可以整个流量切过去，也可以切部分。比如商品系统出问题，可能只需将众多线上集群的一部分流量切过去，就能缓减相当大的压力，这也是一个可行的策略。

## 限流

互联网中除了正常用户以外，还有恶意的用户，包括一堆机器人，比如说友商的机器人，所以我们必须得做限流。但限流一定要谨慎使用。限流策略基本上可以分为4个层次去执行。

**第一层是 Nginx 层限流。**除了基于 Nginx 原有模块自主研发的一套限流策略，我们还有很多限流的规则。

**第二层是 Web 应用层的限流。**规则其实和第一层大致相同，但是因为 Web 应用层主要负责展现逻辑，且它的逻辑特别清楚，所以这一层大致也只能做单一的限流了。

**第三层是业务应用系统的限流。**这一层的限流风险特别大，几乎不希望它做限流，因为业务系统一旦做限流，对用户体验影响特别大。比如我们刚才提及的库存系统，一旦限流，可能导致用户流量流失，尤其是新用户的。所以，轻易不能开启业务应用系统的限流。但可以加缓存，包括本地的、热点的、远程的等等，只要加足够多的缓存就可以解决性能问题。

限流的标准来自于压力测试。比如说做一个调用库存系统，对于交易系统来说每秒调多少次合适？标准设大了不好，设小了也不好。如果设大了就会被压光，如果设小了会影响正常的用户，所以我们又做了反复的线上压力测试和监控，最终将其设为一个比较合理的值。

**第四层是数据库的限流。**无非是限定一分钟或一秒钟之内数据只写多少次，剩余的请求要么直接驳回，要么放在队列里慢慢排队。

## 分流

最早京东用的秒杀是和交易系统结合在一起的，后来被我们分离出去了。对于秒杀系统来说，第一个要做到的就是**数据完全独立部署**。其实每次系统卡，大部分都是在写数据。对于写数据来说，不管前面的系统有多大的分流，多大的集群扩展，它肯定会落到某一台机器上。拿库存来说，它最后总是要落到某一个 Redis 上来，你很难解决这个问题，所以数据要完全物理部署。秒杀系统的特色就是它的热点数据比较集中，每次刷秒杀产品就剩那么几个了，干脆就把单个商品的库存、促销等数据放到一个单独的集群里，性能会好很多，而且最妙的地方

就是不影响正常用户在主交易系统的下单。

**应用部署定制化策略**也是和业务密切相关的。对于京东购物车来说，购物车打开会看到库存状态，会告诉你这个商品的信息，这不影响购物的。万一这个服务出了问题，你还是能看到库存状态，不影响下单。对于购物车库存状态，以及结算运费险等信息，我们通过前面的大节点将其分到单独部署的用户集群。即使整个集群有问题，也只是线上用户反应比较大，对下单量几乎没有影响，而且这样做减小了主业务模型的复杂性。

## 降级

降级是万不得已而为之的。因为线上系统太过复杂，依赖的东西很多，包括软件和硬件，所以首先要做的是**系统级别的降级**。系统级别的降级是什么意思？举例说明一下，比如说成功页如果挂了，用户一点提交系统直接跳到订单中心，在订单中心还是能够看到订单，能找到付款按钮；如果订单中心也挂了，那就直接跳到订单详情页面。

其次是**业务功能模块的降级**。比如说实时价格，并不要求及时，就可以做降级，这个后面会结合实时价格案例去说明。还有一个是**页面级别的降级**，比如说京东首页上有很多动态的东西，需调用后面的大量服务算出结果来。我们准备了一个静态页面，万一动态页面扛不住，先把静态页面给推上去，这样用户看到的信息可能会少一点，但是起码有东西看，不会立即离开。

再次是**非关键业务的屏蔽**。比如之前提到的购物车的状态，为不是关键信息，降级后就可以不再显示。

最后是**远程服务降级到本地缓存**。比如说 POP 商家算运费，一般商家会给我们一个运费模板，包括某个商品商家应该收多少运费，但是因为和 POP 商家联动，所以一旦出现问题我们就会从本地缓存里取出一个默认值暂用。

## 读写分离

第一个读写分离原则是整体系统的读写分离。

第二个是读写分离之后读性能不能集成。读性能提升比较简单，可以用内存缓存、远程缓存来实现。热点数据一般我们放在内存缓存中。比如说京东在 618 时达到 500 万单以上的规模，假设一单是 10 个商品，也不过就是 5000 万个商品，这些对我们内存存储来说完全可以放下。

第三个是写性能提升。一个提升方法是通过京东数据消息队列服务把数据拿出来，再用 Redis 往库里去写；还有一个方法是做数据分片，比如用 Redis 和其他 DB 做分库分表，尽量把集群扩大，把热点数据写进去。

第四个是动静分离。这是现在大家都会的方法，把静态和动态的页面拆分了，静态页面上 CDN、单独部署，不要和动态页面混在一起。对交易的部署结构来说，将购物车静态页面直接放在最前面的大节点上就可以了。

## 线上压力测试

线上压测，主要关注 3 个方面。

第一个方面是读逻辑的线上压测。对于读逻辑来说，线上压测是完全无压力的，特别好实现，因为不会破坏用户的数据，最多把你的系统压出问题来，因为自身可控，系统很快就可以恢复。

第二个方面是写逻辑的线上压测。最难的是写逻辑的线上压测。京东现在线上压测有很多非常重点的写逻辑，并针对不适合降级的写逻辑做了线上压测。如



库存、下单等，背后有大量需要清洗线上数据的工具，都是需要做线上检测的逻辑。

第三个方面是**覆盖范围**。加哪些链接，加哪些业务流，主要取决于：对平时线上业务的理解，哪些量级会影响到主交易流程的业务周期流向；平时要注意分析线上的日志，观察哪些请求调用量比较大。压流过程中要看软件和硬件两个方面的指标，硬件方面包括 CPU、内存、硬盘、网络，因为现在每一个系统都不一样，有的是 CPU 密集型的，有的加了太多的热量缓存，有的是普通的操作硬盘，而有的看重网络连接数，接口流量、出口流量都要观察；软件方面主要吞吐量、并发数、服务响应的时长，因为整个交易系统已完全服务化，服务化后服务端的响应时间是多少，客户端的响应时间是多少，要在压测的过程中进行观测，以找出系统真实瓶颈。比如说，服务端的时间很短，客户端时间很长，就可以检查客户端到服务端之间的网络链路，包括底层架构，是不是有问题。

## 预案

线上出问题不可怕，可怕的是你没有预案。我们现在 618 期间准备了一千多个预案，最后执行了大概有一百多个。

预案的第一作用主要是**发现问题**。有监控平台、日志平台来做软件的监控、硬件的监控、业务方面的监控，还有人工监控。

预案的第二个作用是**定位问题**。系统出问题了，是什么原因产生的，问题出在哪个点，都需要我们结合监控数据和日志一起来分析和定位。

预案的第三个作用就是**解决问题**。方法就是前面提到的那些，无非就是打开路由开关进行分流，或者如实在不清楚就进行降级。

预案最重要的是做出来以后要进行**线上的演练**。如果没在线上演练过，到时出了问题相应的预案很难生效。还要进行压力测试，因为每次压力测试都可以形成一个很好的演练预案的事件，每次压力测试我们总会把系统的瓶颈点给压出

来——当系统处于不健康的状态时，你平时准备的那些预案就可以用上了。

## 应用实例：实时价格系统

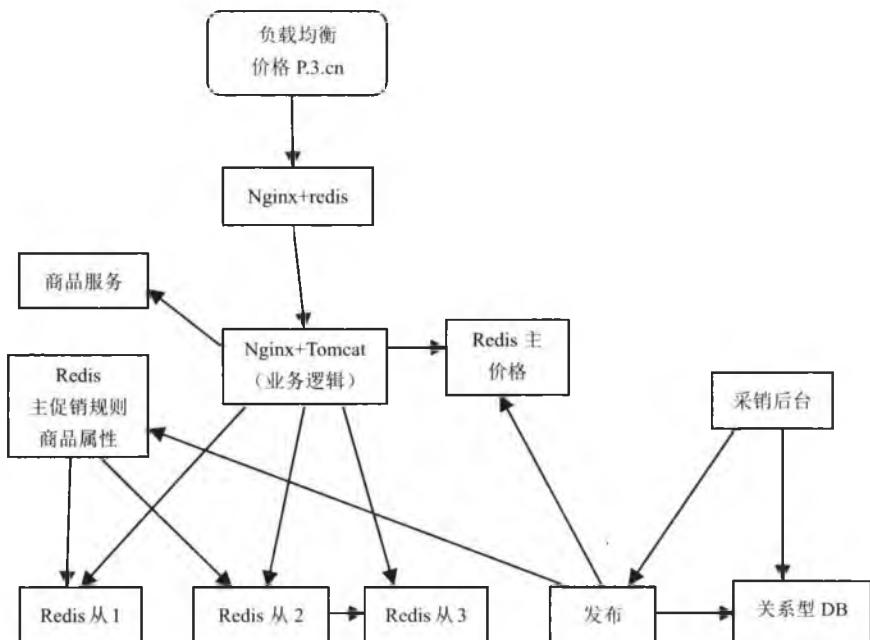
原则是死的，这些原则在各互联网企业的表述也大致趋同，但在电子商务领域如何具体加以应用呢？下面，我们就将京东实时价格系统作为一个实际案例加以分析。



京东戴尔显示器商品网页

在上面这一网页截图中，特意把京东价标出，是想以此商品价格为例加以说明。价格是怎么算出来的呢？既然每个商品都设了基础价格，而且在该商品上做了促销，那么最终价格就是结合基础价格和促销算出来的。

看起来是一个很简单的业务，其实它需要大量系统的支撑。我们先来看一下实时价格系统简化架构图。



在技术方面，我们目前大量使用的是 Nginx+Redis+SSDB。Nginx 的职能有两个，一是业务的路由，二是业务的简单处理。

在业务流程方面，公网上 P.3.cn 就是我们的用户解析，进来就到了我们第一个节点——Nginx+Redis，这个节点是线上一组集群，Redis 里面是存放已经算好的价格，Redis 和 Nginx 布署在同一台机器上，该机器完全可以随时扩展，所以 Redis 里存放着全量的商品价格信息，否则很难使用扩展。这是第一层，几乎放了全量，后面是 Nginx 开拓的业务逻辑层，还有一个就是主价格的 Redis 节点。这一层做什么用呢？一方面，如促销做失效处理，可以到后面重新算一次价格；另一方面，有一些非热点商品即使没有缓存进去也需要回源，由回源调用后面的业务逻辑层，再由业务逻辑层把价格算出来，然后放到主价格的 Redis 主机里面，最后由 Redis 主机云把数据同步到 Nginx+Redis 水平扩展的集群上去。

采销系统把这些价格写到关系数据库里，我们专门有一个异步的发布程序，从关系数据库里把价格提取出来，然后把价格促销信息写到促销的主 Redis 集群里，再把算好的价格信息放到回源服务 Redis 主架构集群里进行同步。因为促销信息实在太重要了，所以仅存放促销信息的 Redis 集群我们就有 4 个。第一个是

主集群，第二个是主集群下用于同步的从集群，其余两个也为从集群。主集群可同步到从二，从二同步到从三，共同预防 Redis 宕机，到时我们有 3 个集群可以定义线上流量，可以很从容地进行灾备切换。从三和从二针对复制，可以减少主 Redis 集群复制的压力，这是后面一个业务的闭环。

实时价格系统应用了如下几个原则。

一是**水平扩展分原则**。这个实时价格系统最大的优点就是无状态，每一个数据都是一模一样的，使用起来很方便。其瓶颈在于主 Redis 的数据负载能力，因为如主 Redis 后面承载的数据分片过多，单独数据一致性比较难保证，性能也就无法保障了。

二是**限流原则**。Nginx 节点加了很多限流规则，还有大量限流数据。整套系统最后做了线上压测，发现整个系统的瓶颈点在于回源的调用。原因是其中的业务逻辑计算太复杂，既要获取促销的信息，还要获取商品的信息，甚至要用大量服务完成逻辑运算，从而导致回源调用困难。

三是**降级原则**。针对回源瓶颈我们会进行降级。进购物车前看到的价格信息可以允许不一致，但要以购物车为准，因为购物车不能降级。这样做对大部分商品而言问题不大。

四是**读写分离原则**。读写逻辑如前文陈述，其实是几个读写业务逻辑的闭环——后面的系统出了问题绝对不会影响到前面的系统。（作者/王晓钟）

# 4 完美风暴——海量订单处理

## OFC 的重要性

2014 年的 618 显得和以往任何店庆促销日都不同，不仅仅是因为电子商务本身在中国不断飞速发展对京东系统带来的挑战，更为重要的是 2014 年 5 月 22 日刚走入美国纳斯达克殿堂的京东聚集了最耀眼的光芒，能不能保持这样的光芒，618 则会是一份很有说服力的答卷，当然我们最终给出了满意的结果。作为一个普通的购物者，当我们在浏览器中输入 `www.jd.com` 并回车，便可以看到京东商城的首页，根据自己的需要选择喜欢的商品，然后加入购物车，提交订单后，即可享受京东的极速物流体验，最终完成一次简单快乐的购物历程。其实，订单提交后，需要经历多个环节和各个系统的处理才能完成使命。其中有一个环节是订单必须经过的，而且这个环节连接了用户下单和订单在库房的生产，就是订单履约中心（OFC，Order Fulfillment Center），本章我们就为各位解密这个部门。

2014 年的 618 后，京东技术团队分享了如何应对店庆日及以往促销活动在技术方面的经验和探索。其中有一讲“电商海量订单处理 OFC 系统的关键技术环节”（见京东技术开放日第一期，<http://dwz.cn/v3qc0>），说的就是这个部门做的事情。

这个部门的职责，用彭青的话讲就是，转换用户订单为各终端系统的生产单，并且按要求送达到相应终端系统。举个例子，就好比我们将采集到的原始食材按照客户的不同口味（不同系统）进行烹制，并且在指定的时间内做好后送到客人（终端系统）那里，整个过程包括订单的拆分转移和订单的下传。其实我们从网站下的订单（也叫原始单）在库房是直接生产不了的，需要经过 OFC 这个环节的处理后，才能到达各个生产系统。由此可见，这个环节必然会有大量数据需要处理，而且还要保证时间。

想必大家知道关于京东的 211、311 等配送方式，如果用户选择不同的配送时间，京东的快递就必须在用户指定的时间送达，而如果下游的生产系统收到订单数据比较晚，就会影响后续的配送时间，最终会影响客户体验。现在订单下传，对接的全国库房近 150 个，需要调用的外部处理订单服务也有近 20 个，而每个系统的处理能力和响应能力又各不同，这就需要我们进行相应的调节流量的配置，这其中只要有一个系统存在问题，就可能会影响订单的下传。

## OFC 的形成

2003 年京东网站创立之后，迎来全国电商的快速发展，京东的业务随之不断增加，导致相应的业务系统也在持续增加。直到 2011 年，随着系统增多及业务的需要，逐渐成长出来一个小团队，负责在各个系统之间进行数据的传输，将客户订单数据传到库房，同时需要将非客户订单，如采购单、供应商、内配单等二十多个业务传输到相应的业务系统，至此 OFC 成形。

初期由于各个系统的不完善，导致数据传输总是存在各种各样的问题，订单总会卡在这一环节，而作为传输环节又必须保证数据能正确传输完毕，这就需要我们必须了解上下游系统的业务及数据处理的过程，只有这样才能知道问题产生的原因，才能知道该如何去处理问题。但是由于上下游的系统是需要经常上线新需求的，我们每天需要及时了解上下游系统业务的处理，才能保证我们的环节没有问题。

那段时间兄弟们真的很辛苦，每天需要处理上千个工单，加班更是常事，以至于一个同事说睡梦中都在处理问题单。由于业务领域划分存在问题，系统边界不明确，导致出现许多莫名其妙的问题，兄弟们干得很苦很累，直到彭青来了之后。彭青带着一个厚厚的黑色全框眼镜，个子不算太高却有个小肚腩，学生族的发型让人感觉很亲切。彭青根据他多年的工作经验和对系统的理解，对这块业务进行重新划分，逐渐将非客户工单的数据传输工作交接给对应系统进行处理，将兄弟们解放出来，开始客户订单处理的新征程。

到 2011 年底的时候，在彭青的带领下我们已经成为了二十多人的团队。为了更进一步拓展在客户订单方面的业务领域，我们主动接手了订单拆分系统和订单转移系统。这两套元老级系统是用 .Net 编写的，由于前辈们大都不在职了，文档也不完善，对于系统内部业务了解的人很少，修改非常难。而此时每天由于系统问题导致的事件单多达上百，也就是说每天运维带来的工作量都是可观的，在这样的情况下，接手这两个系统自然全无阻力。

## 技术的改造

### .Net 到 Java

系统接过来了，第一步要做的事情当然是重写。对技术的选取，根据当时公司技术发展战略以及 Java 的普及，我们选用了 Java。重写过程中需要梳理已有的业务，自然少不了不断和原来系统的人员进行交流，去确认业务流程和技术处理细节。经过一个多月，系统的首次重写总算完成，接下来的工作就是上线了。

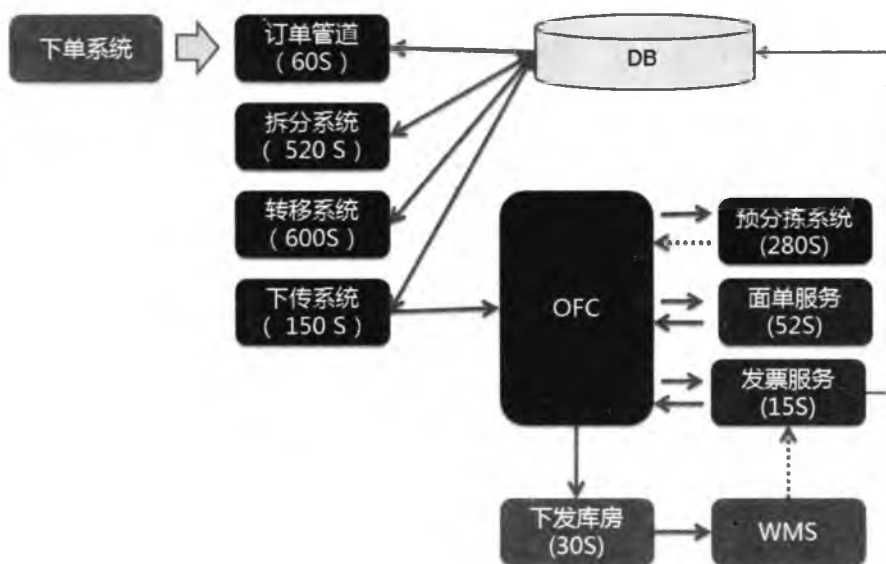
开始小流量地切，我们通过开关进行控制，通过省市县区域分流，到 2012 年 2 月系统算是上线了，而之前的 .Net 系统也逐渐退休了。到这时候，OFC 逐渐根据业务划分为 3 块，第一块是订单拆分，第二块是订单的转移，第三块就是我们前面提到的订单下传和回传。

这里要给大家解释一下：

1. 订单拆分,可能大家想到了,就是将用户下的订单,根据我们库房的不同分布,比如大家电仓、百货仓等进行拆分。当然这是拆分最初的也是最基本的理解,到后面会给大家讲讲现在的拆分是什么样子。
2. 订单的转移则是根据拆分完的订单以及库存等属性,将订单转移到下游的系统。
3. 订单下传和回传主要是指,转移之后的订单调用库房相关的预分解、打包等服务环节,进行订单下传和生产,当订单顺达客户手里,还需要将订单的相关数据进行回传。当然这里只是粗略介绍,后面我们会更详细地给大家呈现。

## 211 订单履约率提升项目

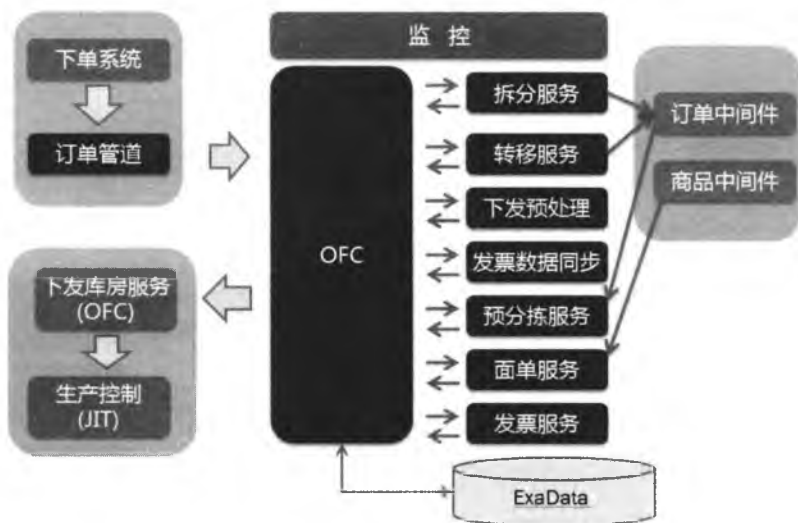
重写完之后,系统总算可以正常运转了,而接下来的事情就是对系统的进一步梳理和优化,以更好地支持将来业务需求的发展以及技术方面的扩展。当然有的时候系统的改进往往是由于外部业务的无法适应导致的,这也符合变革的本质。用户体验至上一直是每一位京东人追求的目标,就连我们的老刘也会隔三差五在网上下单来体验一下,而这次老刘发现了一些问题。当他下单后,等了半天才收到订单,这让老刘无法忍受。经过调查后发现,从下单到库房竟然花了两个多小时。



改造前的系统整体设计图



于是老刘立即成立了一个叫作“211 订单履约率提升”的项目，该项目涉及 11 个系统的升级改造，包括订单交易系统、订单管道系统、拆分系统、转移系统、订单任务系统、OFC 相关系统、预分拣系统、面单系统、增值税资质服务、发票系统、WMS 系统。其中 4 个系统需要全面改造，3 个系统需要大量改造，剩下的 4 个需要少量改造，而且由于与订单相关的业务点多且逻辑复杂，无法在测试环境下全面测试。这不仅影响着整个订单的正常生产，甚至会影响财务相关业务。项目任务非常重要，要求两个月内保证订单从下单到库房的时间缩短到 5 分钟内。大家马上开始了工作——需求讨论 6 天，设计方案 5 天，开发 15 天，功能测试 20 天，性能测试 44 天，上线部署调试 26 天，总计工时达 5066 小时，最终实现了项目目标。与此同时订单下传各环节的服务性能指标也得到了规范，使得订单下传趋于稳定，理顺了订单下传流程。技术方面也得到了锻炼，使用了 Zookeeper 分布式配置、CXF Timeout 设置、Log4j 多 Tomcat 示例配置、Oracle 数据库分区转历史方案，数据库使用了 Oracle Exadata、MySQL。在这过程中和 Oracle 技术团队直接沟通多达 10 次，在数据库设计方面、性能调优、转历史数据方面都得到了提升。更为重要的是锻炼了团队，对于战胜艰巨任务有了更大的信心。下面是系统的整体设计图。



改造后的系统整体设计图

对于拆分，在几位大牛对系统的业务进行梳理后，发现部分业务有些混乱，业务领域划分得不是很清楚，拆分系统中除了需要根据商品的不同属性进行拆分外，还需要对订单中使用的金额、优惠、运费等信息进行分摊处理。这几位大牛

敏锐地发现系统这样设计是有问题的，于是就把金额信息处理逻辑拿出来，专门做成一个服务——OCS 订单金额计算服务（OCS），拆分只需要对其调用就可以。同时，我们对 OCS 分摊结果的数据进行了持久化数据存储。系统这样设计，不仅解耦了拆分服务之前混乱的业务处理逻辑，OCS 的数据也一举填补了公司这方面数据的空白，成为其他系统使用和处理业务逻辑的数据基础来源。到现在为止，直接使用 OCS 数据的系统就有二十多个，其重要性不言而喻。

## SOP 合页单项目

2013 年，公司级项目 SOP 合页单要启动，即用户购物车里既有京东自营的商品同时有 POP 商家的商品（SOP）。在结算的时候只需要提交一次（之前只能分开提交，类似淘宝多商家的商品只能单独提交）。为了改善用户体验，同时需要将提交之后拆分完的子单结果显示出来，需要我们团队提供一个拆分服务供交易组使用，这是一个重大的考验。下单环节的速度非常快，TP99 一般都是几十毫秒，而我们目前的服务则是几十秒，完全不在一个数量级。为了保证项目能够顺利完成，我们既需要满足日常的业务需求，同时要新切出一个分支进行修改，用于此次项目，同时需要将针对新需求的代码及时同步到这两个分支上，任务非常艰巨。解决了开发问题后，就要想着如何在性能上有所提升，比如，可以放在内存里处理的就放在内存中操作；尽量减少对外部服务的依赖；对于非同步化的操作进行异步化；对于部分服务我们甚至采用降级的方式，在必要时通过开关进行降级，保证整个服务的整体性能。如此这般后，我们主动要求性能测试组对我们的服务进行性能测试，在代码级别进行了优化，最后在指定的时间内成功地完成项目。而此时我们在维护着同样级别的 3 份拆分服务代码，老的下单对应的我们前面说的老拆分，新的下单对应的我们新的拆分，还有我们为交易系统提供的预拆分。

而在此时最困扰我们的不是维护这些系统，而是经常会由于网络不好，使一个订单的服务超时，进而导致服务进行重试，而事实上订单已经提交成功。这就可能使我们错误地提交两次甚至是多次订单，比如客户下一个原始单，需要拆分成两单，但是由于上述原因可能会得到多单；如果用户选择货到付款，会给用户造成困扰，会带来配送的成本，如果是在线支付的话则会导致公司的损失。刚开始的时候没有解决方案，只能通过监控去发现，发现后人为锁定这些订单，而这样不但增加了运维压力，而且人工处理难免会有失误。由于我们在提交子单之前会获取订单号，每一次获取的订单号都是新的，这会导致调用这个服务时对订单号是无法防重的。后来海波想到一个防重方案，就是我们在调用这个服务之前将订单号信息输入自己的防重库，新订单来的时候先在防重库中进行查储，如果有

订单信息则说明之前提交过，本次提交失败，然后直接把库里相同订单号的数据拿出来提交即可，这样还可以节省订单号。如果库里没有查到，我们将该订单号插入库中，同时调用服务。问题得到有效解决。本次提交经过这一系列的处理优化，系统总算是比较稳定了。

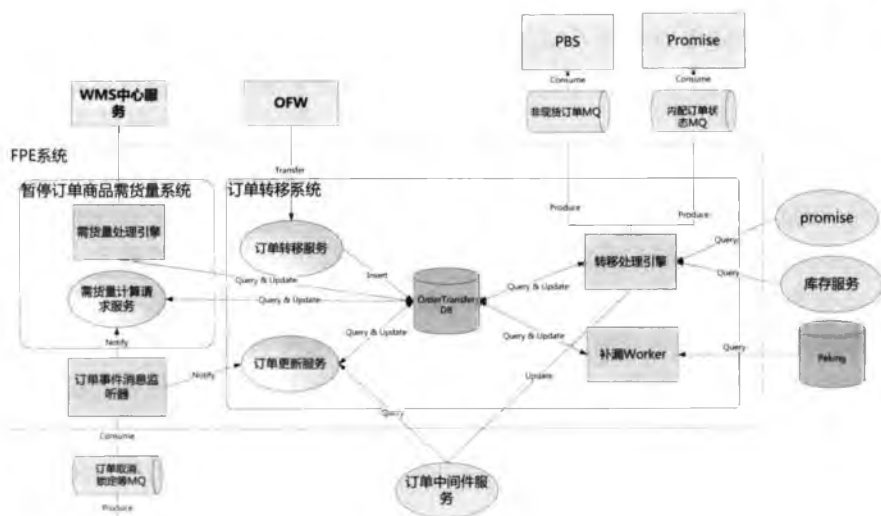
## 转移架构升级

转移系统也进行了大规模的调整，为了进一步保证订单及时准确地转移到下游的库房系统，转移团队在业务和技术架构上进行了一系列的改进：业务和数据处理异步化，即将可以异步化处理的业务和数据放入分布式队列，由对应的模块处理；使主流程业务简单快速流转；数据处理并行化，将数据切割成多个业务单元，并行处理业务单元；针对变化少、实时性要求不严格的热点数据，使用缓存并配以更新机制，以提高性能；对于业务洪峰，通过平滑控制保护后续系统不被洪峰压垮。

在业务流程方面也进行了优化。由于涉及订单生产流程，需求变化速度非常快，需要不断梳理现有流程，去除不必要的流程，减少有新需求时对不必要业务流程和分支的考虑。同时，需要对现有分散业务不断地抽象和改造，以方便业务扩展。

面对这么多的优化和改进，每次上线的风险无疑是巨大的，如何规避风险呢？那就是要分流、可配置化，以及运营工具先行。由于新上线的项目风险较高，特别是容易忽视一些对外交互的小功能，而发生线上问题时又无法及时切换。因此，需要对业务上线进行分流，并且通过灵活便捷的配置中心随时进行控制。对于异常情况一定要优先考虑，并且开发相应运营工具，以备紧急情况使用。尤其不能抱有侥幸心理，认为小概率事件不会发生在自己身上。

转移团队的负责人铁总（大家总是这样称呼他）已经从事电商十余年了。这个来自湘西的汉子对待工作总是严肃认真，但面对生活却又充满热情；结婚前总会泡在游戏中，或者痴迷羽毛球，女儿出生后便成为了他的一切。在谈到转移未来的规划和发展时，他充满自信地说：“将来会在保证客户体验的同时，更多地通过在成本和流程上优化来降低成本。库存分配将在保证订单履约的前提下，打破现在先下单先占库存的规则，提高商品库存周转率和现货率，同时给客户提供更早的收货时间选择”。



转移系统整体流程图

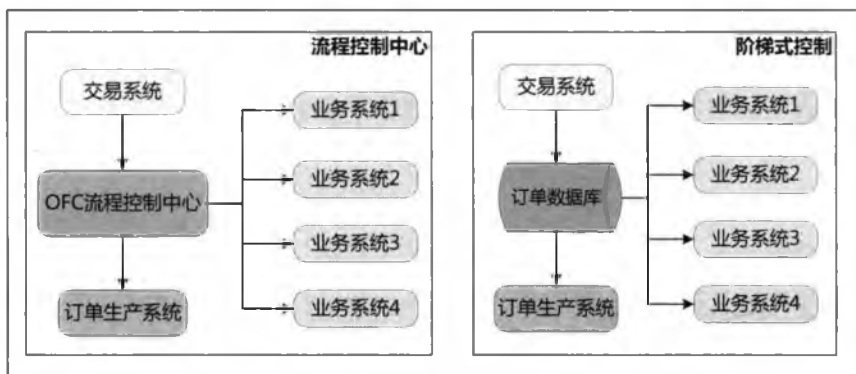
## 不得不爱的运维

刚开始负责客户订单系统时，每天要处理上千条 Ticket（订单事件），而现在只需处理几十条。这种锐减，不仅说明了系统日渐健康、业务逐渐规范，更证明了我们的运维流程和运维制度正日趋成熟。这些成果都离不开善于分析总结的文杰，他是一位来自山东的 80 后，在团队中主要负责运营流程优化和与协调相关的工作，团队在运维方面的问题目前还没有他解决不了的。OFC 是连接用户和全国终端库房的重要的通道枢纽，这其中的任何系统出了问题，都会导致订单无法正确实时地到达终端库房，后果都是不堪设想的。因此，每新增加一个库房都需要团队进行库房的终端系统部署和调试，直至生产系统测试完成为止，我们称此为开仓过程。随着公司不断发展壮大，订单业务不断完善，全国现存仓库已超过 150 个，这都是文杰和团队无数日夜的付出换来的。支持审计也是不可忽略的一部分工作，每季度我们都会给同事讲解新业务，给他们解释差异订单的原因。同时，我们还负责新业务的学习推广，让团队的新成员能够快速了解业务知识、熟

悉业务系统。伴随着业务和系统的日渐完善，我们也在不断地尝试和推广系统的智能化运维与支持，相信在不久的将来我们一定会实现无人化系统运维。

## 从 618 到双 11

从 2012 年开始，店庆促销活动力度在逐次增加，订单量则成倍增长。伴随着订单量的增加，系统面临的挑战与日俱增——订单业务越来越繁杂，业务处理流程也越来越多，很容易出现数据不一致问题。因此，在处理海量订单时保障数据一致性非常关键。系统整体控制上要采用流程控制中心，而不是阶梯式控制。之前由于直接依赖数据库，数据库最终会成为影响订单处理的瓶颈，数据的一致性很难得到保证，而采用流程控制中心模式则可以大大减少数据不一致发生的几率，同时可以借助工作流和状态机实现中心控制，这样既便于运营，又方便及时发现和解决问题单据。



流程控制中心和阶梯式控制

## 支持海量订单处理

无论系统如何优化，单个系统总有瓶颈，要支持不断增长的订单处理量，关键在于提高系统的扩展能力。首先，核心系统的每一层都要有扩展能力，可以以实例为单位进行扩展，也可以以集群为单位进行扩展。其次，系统整体要有扩展能力，可以根据实际业务特点，从业务上进行垂直拆分以实现扩展，也可以通过分布式部署来方便地增加一个具备整体功能的集群，从而快速增加处理能力。这相比仅做备份系统而言，节约了成本。

所有核心的 OFC 订单处理系统已实现了水平扩展能力，部分系统实现了分布式部署改造。在 2014 年 618 大促前，正是由于系统具备这种扩展能力，能够在非常短的时间内扩展了处理能力，保障了大促的顺利开展。我们的最终目标

是，所有核心系统都要完成分布式部署。

### 解决数据一致性问题

早期的订单处理流程分散到多个应用系统中，数据来源不统一，也缺乏统一的状态机控制，经常出现数据不一致问题。但同时，也不可能由一个系统来管理所有的流程，因为维护和管理工作会非常庞杂。解决办法是，梳理出订单处理的主流程和状态机，然后由主流程系统负责整体流程的调度和数据的推送。这个主流程可能跨大的业务域，如物流领域和资金领域，每个领域内可以有工作流，但不能与主流程冲突。识别出主流程系统还有其他的优点：一是可只重点建设主流程相关系统，使其成为稳定的系统集群，而非主流系统则可以投入较少的成本，从而既有利于保障业务顺利开展，又能降低整体建设成本；二是主流程系统可以有效地保障生产计划的执行；三是主流程系统可调节系统流量，有效地平滑业务高峰，保护主流程相关各主要系统的稳定运行。

### 支撑运营工作

对运营工作的支持，包括抢险、预防，以及“治理+预防的升级”。在早期阶段，系统架构主要是支撑业务功能的实现，没有为运营而设计，线上系统会因为各种意外而影响业务，让系统团队疲于应付。后来，确立了为可运营而设计的理念和原则，设计时必须考虑可监控、可运营，同时把可用性、稳定性、健壮性等列为设计的重点，在实践过程中确立了自己的方法论。

第一，对系统进行梳理，识别出核心系统，把核心系统建设成为可用性高、可靠性高的系统，保障这些系统少出问题，出问题后系统要能自动恢复。

第二，保证系统出现问题后能快速发现问题，甚至在问题发生前发出报警。为此需要有对数据积压量趋势的监控，以及在有积压情况下吞吐能力的监控。这些监控需要及时，我们针对分布式系统，开发了分布式监控系统，能够迅速地反应每个部署的每一个实例的情况，又能收集整体的运行情况。

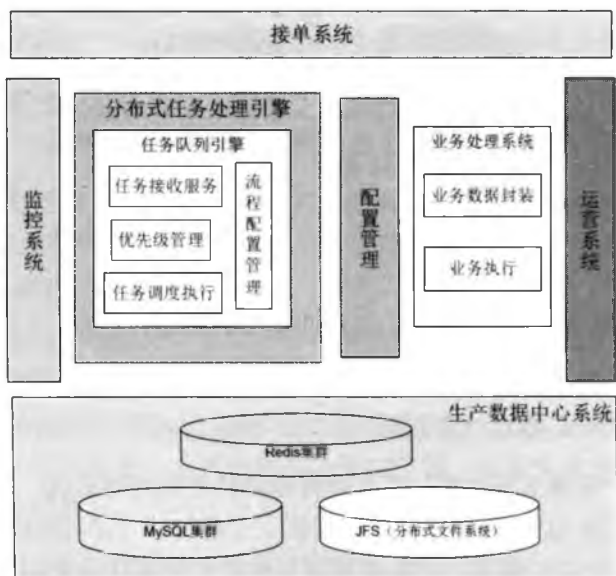
第三，保证发现问题后可以快速定位和处理。为此我们设计了集系统处理能力、数据积压情况、数据处理情况、日志、系统负载于一体的统合分析工具。

第四，一个系统出现问题，往往会将影响传递到其他系统，系统治理势在必行，目前我们已有 SOA 治理平台（正在优化过程中），目标是能够理清各系统的血缘关系，完善 SLA 体系；在出现问题后可以及时评估出受影响的系统，快速做出应急响应。

## 海量数据的开始

### 总原则

订单处理系统与交易系统本身是存在区别的。交易系统直接面对客户，所以系统可用性要求和性能要求非常高，特别是在高并发情况下的系统表现，所以交易系统在架构上的重点在于解决这两个方面的问题。而订单处理则不同，系统短时间不可用，响应出现延迟不会对客户造成直接影响，也就说我们关心的是平均值而不是某时刻的峰值。订单处理系统架构设计的关键在于如何处理海量数据，以及数据一致性的保障。近年来，京东的业务领域不断拓展，订单量飞速增加，所以必须保障系统吞吐能力得到提升。与此同时由于涉及的系统众多，各系统业务处理方式和流程不同，导致各系统性能指标差异较大，所以要定义好各个系统的 SLA 指标。由于订单的业务越来越复杂，那么系统的业务流程也会越来越多，这就需要我们划分好主次业务流程以及优先级，同时需要设计灵活多样的降级方案，保证主业务正常运营。



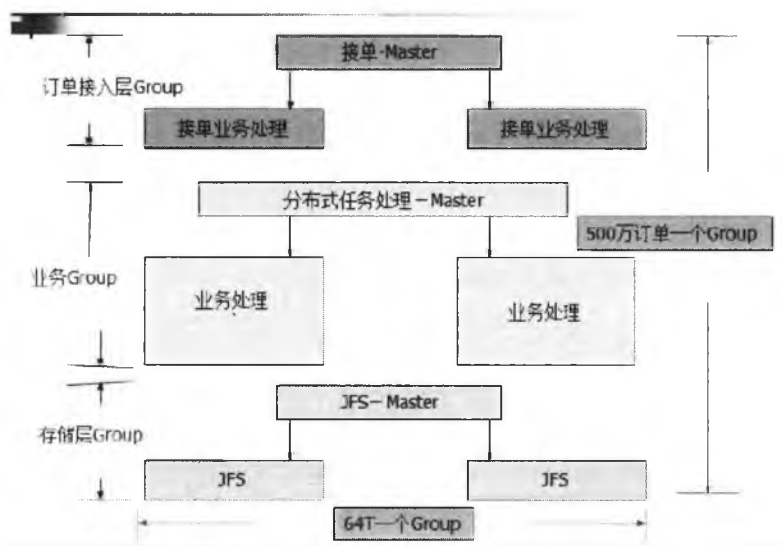
OFC 整体架构图

## 系统保护

涉及 OFC 调用的订单系统很多，而各个系统处理的能力均不相同，不是所有的系统都要承担高峰值处理的压力。这就需要有针对性地控制、调用这些系统，具备削峰和流量控制的功能，以间接保护上下游系统，防止调用方的系统雪崩式挂掉。还有就是监控，要有统一的产能监控；要防止过载，在过载之前要能进行控制，要保证自身系统的安全稳定，还可以采用快速拒绝的机制。

## 分布式系统

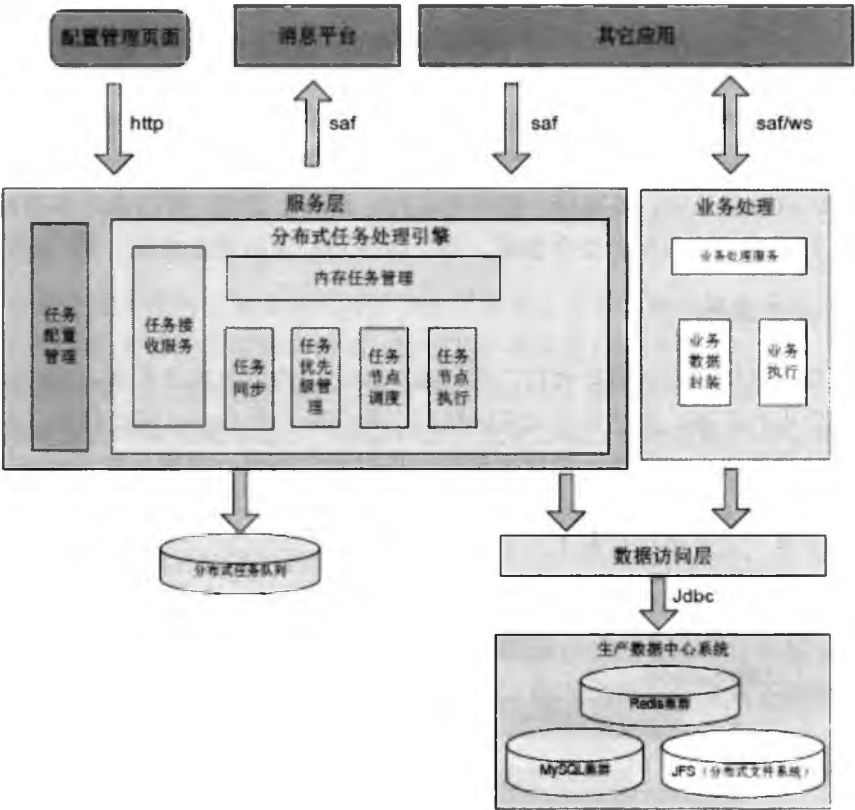
主要是在扩展方面进行设计，保证系统每个切片可以水平扩展，也可以以集群为单位进行扩展，实现分布式任务队列。我们每一个 Group 能处理的订单量在可控制的范围内，一旦某一处出现瓶颈，可以随时部署一个或一套 Group。下图中不同 Group 可以彼此独立部署，也可以整体部署，当某一处出现问题时可以单独进行部署，或者整体流量大时也可以复制部署。



分布式系统架构

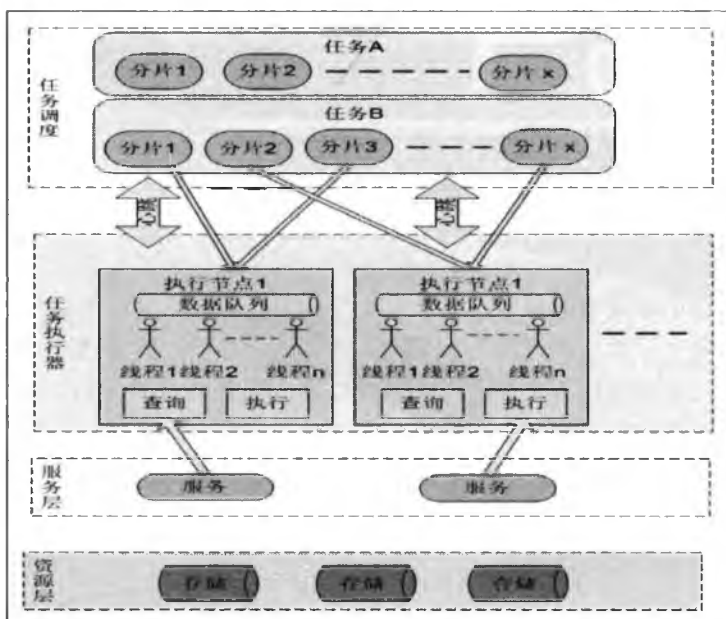
而分布式任务处理架构图如下，在分布式任务处理引擎的基础上，我们可以通过在图下方左侧的分布式任务队列（Redis 缓存）中取任务，然后由我们的引擎一步一步去调度相应的服务，将结果返回给对应的服务进行业务处理，同时也返回我们需要的结果，真正将交易快照数据转换为生产单据，最后将数据推送到客户端系统，比如库房系统、POP 商家系统。





分布式任务处理架构

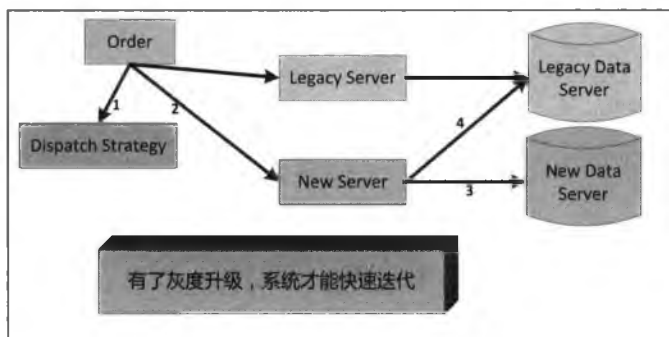
分布式任务队列设计可以通过下图说明。首先我们会对任务进行分片，定义每一个任务为一片，然后将任务在任务执行器中加以处理，而每个任务执行器又有多个线程去处理和调用不同的服务，最终再将结果返回。这里还会有一个异常任务队列，对于那些处理失败的任务，会将其放入异常任务执行队列，进行异常处理流程的执行。可能大家会想，如果系统挂了，那个任务的数据也将丢失，没有执行完任务怎么办。事实上只要我们的原始订单数据存在，完全可以恢复为再次执行，最多执行会缓慢，但不会导致数据一致性出现问题。



分布式任务队列设计

我们的分布式任务队列采用了工作流的机制，支持灵活的流程配置，这主要通过 Zookeeper 来进行分布式配置，可以动态添加业务处理环节。同时，可以自动调节系统的吞吐量，当任何一个环节出现问题时，都会进行自动降速，当问题得以解决后，我们会进行自动增速，保证系统的吞吐量；我们还可以通过配置对一些高级别的订单进行优先生成处理。

系统部署如下图所示：



系统部署图

## 一直在路上的 OFC 人

从 2011 年 6 月初只有五六个人的小组，到今天三十多人的团队，从开始只负责非客户订单相关业务系统，到今天负责公司核心的 0 级订单系统，OFC 业务范围横跨了整个订单生产过程的大半个生命周期。是的，就是这些普通人，在纷乱中不迷茫，在欢笑中不忘使命，表面屌丝内心渴望前进的 OFC 人，义无反顾地坚持在订单系统的一线上。回望着走过的路，我们竟如此平静而淡定，遥望远方，我们依旧充满激情，依旧会绽放阳光般的笑容。因为这是个伟大的时代，伟大的国度，因为这里有一群伟大的人在做着一件伟大的事情——为了让购物变得简单快乐，而我们便是那群人。



OFC 开发团队及测试团队合影

（作者/彭青）

# 产品演进

一个电子商务购物网站到底是由哪些系统构成的呢？这是每一个IT人都想了解的。

本篇就为您揭秘京东整个核心业务链条的各个系统：商家开放平台、仓储、物流、售后、财务、内部办公等；首次详尽披露大家最想了解的两个问题——京东网站是怎么一步步发展演变的、京东和易迅拍拍怎么整合的。

- 商城网站发展史
- 与腾讯ECC的融合
- POP传奇
- 极致体验的供应链
- 售后系统
- 财务系统演化史
- 内部信息化进程



# 5

## 大时代——商城网

### 站发展史

#### 青铜时代

2003 年，非典肆虐京城，中关村所有实体店都是门可罗雀，京东也不例外。当时，老刘为了员工的安全，决定关闭所有门店应对非典。可门店关闭后，京东的资金最多能支撑半年，公司如何经营是一个大问题。偶然的机会，老刘发现通过网上论坛可以直接向客户销售商品，而且不必与客户接触，能避免非典的影响。就是这样，京东开始了网络销售之路。

经过一段时间的尝试，线上销售额持续增长，网络销售为公司打开了新天地。老刘经过慎重考虑决定关闭线下实体店，专营网上销售。而网络销售不能单单依靠公用的论坛，还是要建立京东自己的线上销售平台。这个平台将成为京东未来开展业务的最主要渠道，它也被誉为京东的生命线。

第一版网站的开发还有一段传奇的故事。2003 年老刘开始尝试通过论坛进行销售，他的销售方法与普通人很不一样。让用户先试用，试用好了再买，试用不好不要钱。而在一次线下试用会上，故事关键人物——曹鹏出现了。当时，曹鹏在一家外企做程序员，对光盘产品有比较大的需求。来参加老刘的试用会也是想

看看京东的货到底如何，经过试用，还真是正品行货而且价格便宜。就这样曹鹏与京东结缘，也与老刘熟悉起来。

2003年老刘打算自建网站的时候就想到了曹鹏，当时除了曹鹏还邀请了另外一个程序员。据曹鹏回忆，当时就他们两个人开发这个网站，用ASP技术。

2004年初，网站正式上线，名为“京东多媒体网”。网站的功能比较简单实现了商品陈列，用户可以顺利下单。而下单后，则需要人工打印单子到库房提货，操作比较原始。无论如何，这是京东第一个自主研发的网络销售平台，是京东电商事业的重要里程碑。利用这个平台，用户实现了自主线上购买，京东与用户的距离更近了一步。网站上线后，销售额不断扩大，增长速度惊人。

从2004年到2006年，负责网站开发和管理的人员不足5人。所有与系统相关的工作都由这几个人合力完成。从论坛到后来自建网站，老刘对用户反馈异常关注，他也一直保持着与用户密切沟通、回复用户留言的习惯。因此，老刘特别重视网站的开发工作。据老员工回忆，当时他在公司有两处工位，其中之一就在研发部。平时工作中，他也经常指导网站的设计、文案的撰写等。

2005年上半年，老刘力排众议关闭线下的12家门店，专攻网络销售。网站销售额以每年200%多的速度增长。到2007年6月，网站日订单处理量已突破3000单。

2007年8月京东拿到第一笔风险投资，企业发展进入快车道。订单量的增长也对系统提出了巨大挑战。京东急需找到一位优秀的技术人才来领导团队，为业务增长打好基础。

2008年5月，Donny加入京东，担任京东研发部负责人。进入京东后，Donny仔细研究了现有系统，认为现有系统无法满足京东未来的业务增长，必须进行架构升级，从ASP转向.Net。同时，借此机会对前台页面进行改版，提升用户体验。

## 白银时代

2008年8月20日，一辆面包车停在了北京通州区某个高档小区的一栋三层

别墅门前，从车上下来了3个年轻人，他们好奇地四周张望着。别墅里的人很快出来了，一边向新来的年轻人打招呼，一边从面包车上卸下几台电脑和显示器，还有几个路由器和插线板，有说有笑地搬进了屋。京东未来的网站架构，就诞生在这座三层别墅当中。

在3个年轻人中，有一个刚刚入职3天的小伙子 Frank，他从搜狐离职来到京东，希望在电子商务大潮中施展自己的才华，不过没想到的是，加入京东后的第一项任务，就是收拾行囊投入为期将近3个月的封闭开发。

本次封闭开发的总指挥是负责技术研发的 Donny，大家都亲切地叫他老李。老李是在2008年6月加入京东的，是京东第二个副总裁。老李是典型的南方人，个子不高但透着精明，语速不快但抑扬顿挫。我们都觉得，他的演讲感染力和马云有的一比。老李按照支撑百亿销售额的目标，提出了京东网站重构的建议。在过完618店庆之后，就带领了一支小分队，奔赴通州进行封闭式开发。

这次封闭开发的前期大概有14人左右，后期进入攻坚阶段，又陆续有一些同事过来支援。前期主要是1名产品经理、2名前端开发、2名设计师、1名测试人员及8名左右的研发人员。封闭开发的条件还是蛮艰苦的，虽然是一栋别墅，但里面只有两个简单的沙发和在三楼的一张床，因为老李年长，所以床自然让给了老李，其他年轻人打地铺。8月还好，后来到了10月天气渐渐冷了，有时阴雨连绵，又没有暖气，大家就陆续拿来了更厚的被子。每天作息大概是早10点到凌晨1~2点的样子，好在那时年轻人比较多，应付起来没什么问题。不过有一件比较头疼的事情是，有个别同事鼾声比较厉害，他如果先入睡其他人只有醒着的份了，所以大家强烈要求他加班到更晚，以便其他人能先入睡，保证第二天的战斗力。伙食方面还不错，请了一个阿姨来做一日三餐，味道可口加上僧多肉少，吃起来还是很香的。最惬意的事情，还要数午饭和晚饭后，在这个高档别墅区溜上一圈。

封闭开发的过程就不细表了，有遇到难题的攻关，有面红耳赤的讨论，有痛苦的推倒重来，也有过程中的欢乐和喜悦。项目到后期，时间比较紧张，全新的底层系统联调和测试过程中，会遇到各种各样的问题，眼看项目时间所剩不多，大家都拼命地往前赶工。这个阶段老刘还过来看望了大家，并请大家到附近的一个饭店聚了聚餐，给大家鼓鼓劲，那顿饭记不得味道了，但大家心理压力都蛮大的，吃得也并不轻松。

3个月后，经过大家的努力拼搏，封闭开发顺利结束了。新版京东商城网站在11月初顺利上线。在与用户互动的过程中，虽然发现了一些 Bug 和性能问题，但经过修复，用户总体满意度和转化率还是有比较大提升的。在这次改版中，网

站页面比较大的改动有4处：第一是商品分类不再采用全部罗列的方式，而是改为二级分类弹层模式；第二是增加商品排行榜；第三是弱化了登录，节省空间展示更多商品；第四是购物车展示在首页，方便用户随时结算。

新版网站为京东在2008—2010年的快速发展打下了坚实基础，订单量从日均几千单增长到十几万单。同时，经过3个月的磨合，一个崭新的研发团队也随即诞生。这支队伍的成员也都逐渐成长为京东技术的骨干力量。

## 黄金时代

关注用户体验是京东的传统。从老刘到一线员工，每一个人心里都把用户放在首位，一切以用户的利益为重。

经过2008年的改版，京东网站的性能有了大幅提升。而网站的各项体验还需要不断完善。2009年，由老刘提议，在京东网站上开辟“总裁在线”版块。老刘及多位高管定期在网上回答网友的提问，并快速帮助网友解决问题。

“总裁在线”功能上线后，得到了广大网友的积极响应。老刘和公司的几位高管也确实在线帮助网友。后来，随着商城网站功能的逐步完善，以及服务水平的不断提升，京东专业的客服团队是可以帮助用户解决各项问题，“总裁在线”也就成功隐退。

在京东发展初期，关于物流的咨询是最多的。但是随着京东物流服务的提升，咨询量也逐步下降。而在这个过程中，“订单跟踪”系统起到至关重要的作用。

### 我的包裹在哪

2008年，京东的物流体系也正经历快速的发展，逐步构建起竞争壁垒，京东送货快的口碑在一线城市流传起来。不光送货要快，老刘还希望把货物流动的信息清清楚楚地显示在用户面前，打消用户的顾虑，让用户放心地购买，安心地等待收货。



2010 年在老刘的提议下，负责订单展示的产品经理开始着手设计“订单跟踪”功能。这个功能很简单，就是把从用户下单开始的整个服务环节的关键节点都记录并显示出来，无论用户何时打开网页，都能看到自己的商品现在在哪，是什么状态，哪个配送员负责，他的电话是多少。

从页面内容来看，这是一个非常简单的信息展示，但其背后的实现机制并不简单，这需要从用户下单那一刻起，京东信息系统就把商品的所有流转都记录下来，这需要订单系统、仓储系统、分拨系统、配送系统、站点管理系统、配送员信息系统等各个系统的打通和联动，才能最终将每一条信息完整、准确地显示给用户。

京东做到了。订单跟踪系统刚刚上线，就获得了用户的一致认可，客服电话呼入率大幅降低，用户满意度大幅提高。这种“一切尽在掌握”的感觉对用户来说是极好的。在电商需要被更多人接受和信任的年代，这种服务的确直击痛点，并很快成为各大电商效仿的对象。虽然现在“订单跟踪”已经成为电商的标配，但京东是最先推出该功能的。这种对用户体验的关注和创新精神，一直推动着京东的快速前进。

到了 2012 年，随着仓储配送系统的加强、流程制度的完善及大数据的积累，京东进而推出订单时效 Promise 系统——根据商品库存状态、仓库所在地、收货地、下单时间、配送能力等因素，提前预测订单送达时间，让用户更早地知道何时能收到货。比如，在周五上午 11 点前下单，北京的用户会看到“预计下午 6 点前送达”的提示，这样用户就可以放心在单位收货，而不用纠结到底是送到单位还是送到家里。订单 Promise 从服务角度来讲，是京东给用户的购物承诺，当有送货延迟，会给用户一定补偿。同时大数据团队也会反复测算数据，找到问题出在哪里，如何更精准的预测。京东靠着对自己严苛的要求，不断提升着用户体验。“京东送货快”，就是这样一点一滴形成用户认知和良好口碑的。

不光是在物流服务中，京东利用大数据分析的方法。在用户浏览网站时，我们也要通过数据分析用户的习惯，从而指导网站设计。而如何收集用户浏览习惯的真实数据呢？

## 建立用户体验室

2006 年年底，一本书悄然在互联网圈，尤其是产品设计圈流行起来，它就是 *Don't make me think*。此后，用户体验更多地被重视和传播开来，什么是好的用户体验，怎样做出好的用户体验，每个互联网公司都在思考，京东也不例外。

做好用户体验，就是要多和用户接触。老刘在这方面已经给京东的产品经理树立好榜样，经常亲自参与网友见面会、接客服电话、在线交流等。而要想和用户在一起，首先必须能快速找到用户。对此，京东的产品经理们马上想到一个地方——京东自提点。在京东发展的初期，用户还抱有试试看的心态在京东购物，一手拿货一手交钱的习惯让不少人选择了来京东自提点先提货后付款的购物方式。这里有新用户也有老用户，不同性别、年龄的用户都可以找到，而且地点固定、客流量较大，所以在早期，京东自提点就是京东产品经理们做用户调研和访谈的最佳场所。

在 2009 和 2010 年，京东的办公地址在北京海淀区苏州街附近，同时那里也建立了全国最大的自提中心，自提中心内配备有两台电脑，供用户查询订单或挑选商品。用户在操作时，产品经理就会悄悄站在用户的身后，观察用户怎么使用京东的网站，在哪里停顿了，在哪里出现了操作错误，并观测用户的反应和后续行为，这种背后的观察是最真实自然的。自提点的用户调研，让产品经理们可以随时获得第一手资料，不断优化网站体验。

到了 2011 年年初，产品经理和设计师们觉得总在用户背后观察也比较受局限，我们需要有更好的环境，与用户做更深入的交流或测试，所以在公司的大力支持下，京东研发部开始筹划建立京东用户体验室。用户体验室主要由 3 个房间组成，分别是访谈间、测试间和观察间。访谈间主要是一张长方形的桌子，用来对用户进行焦点小组访谈，大概 6~8 个用户围坐在一起，由主持人引领进行讨论。测试间环境相对轻松，配备书柜、沙发、绿植等，主要设备是两台电脑和一台价值不菲的眼动仪，主要用来记录用户的鼠标操作和眼动数据。观察间很有意思，外部是一个单面镜。什么是单面镜呢？就是从外面看它是一面普通的镜子，但是镜子后面的人可以透过镜子看到外面的景象，所以往往产品经理会坐在观察间，观看用户的操作，同时用户说的话、操作的界面，都会通过麦克风和显示屏传送到观察间内，非常适于进行实时观测。

这里要特别介绍一下眼动仪设备。京东的眼动仪是瑞典 TOBII 公司生产的 X120，通过图像传感器采集的角膜反射模式，该眼动仪可以计算出眼球的位置和注视方向，结合精密而复杂的图像处理技术和算法，可以构建出一个遍布注视点的平面参考图。有了这个设备，我们就能知道用户看了哪里，注视了多久，点击了哪里等信息，并通过数据分析，得出非常精确量化的结论，辅助产品经理优化设计方案。

用户体验室的建立为京东在 2012 年的改版奠定了基础。2011 年下半年京东启动前台网页改版。本次改版的核心宗旨是展现京东综合商城的形象，同时提升美观性，吸引更多女性用户的关注。

新版网站上线后，通过数据对比发现，首页转化率提升了 4.74%。用户体验进一步改善。

## 不避讳差评的公司

2012 年改版后，京东的品牌知名度得到了大幅提升，越来越多的人开始关注京东。网站上一点小小的瑕疵，就可能影响大批用户的购物体验。京东的产品经理也越来越小心地对待每一次改进。就拿我们的评价系统来说，历史悠久又不断革新。

可以毫不夸张地说，京东是商品评价做得最早，也是做得最好的电商公司。为什么京东这么关注用户的评价，还要从老刘说起。老刘是出了名的喜欢与用户互动，直到今天，他还一直坚持每年亲自送一次货，拉近与用户的距离。在老刘和京东人看来，信任用户、尊重用户、让用户帮助用户是最高效的缩短与用户间距离的方式。京东的商品评价能做得如此成功，主要有几方面的因素。

首先，京东最早就在 BBS 上进行交流和销售，社区氛围浓厚，从之前论坛一路跟随而来的忠实用户，以及硬件商品的发烧友，本身就具有非常强烈的分享意识，这批种子用户质量非常高，写作的内容很有价值。就像滚雪球一样，高质量的评价引来更多人参与，而更多人参与也给了写作者更多的激励和荣誉感。那时候的京东不像一个平台，更像一个硬件爱好者的大社区。

其次，京东评价有一个规定，只有购买过这个商品的用户，才能发表评价。虽然这会阻止很多想说两句但又没有购买过的用户，但这样做是非常合理的，购买且使用过这个商品的人，最有发言权，评价的内容最可信，那么这些评价也就更有价值，更能够帮助其他用户进行购买决策。良币驱逐劣币，形成良性循环——得到帮助的用户，在购买后也更愿意帮助别人，把获得的帮助进行传递。

再次，京东还非常精细地使用积分进行引导。比如，评价不同品类的商品获得的积分数量不同：前5条评价可以获得多倍积分；会员级别越高，获得的积分也越多；高质量的评价可以获得额外积分等。通过这些刺激，对用户进行引导，让用户非常乐于贡献高质量的评价，同时又得到回馈。

很快，京东的商品评价就成了京东宝贵的财富，百度对京东评价的收录量巨大，权重也很大，为京东带来大量用户。很多用户购物之前，都会来京东反复查看评价，到后来也会以评价的数量来代表商品的热销程度，作为购买决策的依据。

这还不够，京东的评价还分为“好评”、“中评”、“差评”3个Tab页，用户可以方便地查看该商品的全部“差评”。这会不会让用户放弃购买？起初产品经理们也有这样的担心，但在老刘的坚持下，最后还是坚持这样做。通过数据观察，总的订单量并没有受到影响。用户发现商品的不足，可以更全面地进行对比，最终做出一个最优的购买方案，反而降低了退货和投诉，这是我们最惊喜地发现。同时，对于差评较多的商品，我们会有专人巡视，考虑是否下架处理，与供货商进行交涉甚至是处罚。这就是用市场手段优胜劣汰的最好案例，也是京东深谙商业本质的最好体现。

再后来，我们还进行了一项更有意思的实验，通过展现不同的评价以及不同的排序，进行A/B测试，找到给用户最大帮助、转化率最高的那套内容组合，以此来优化。有3个团队提供不同的算法参加了这次“竞赛”，这是一个非常有意思也有价值的比赛，各个团队可以不断优化自己的算法。比赛大概持续了3个多月，最终成都研究院团队胜出，新的评价筛选和排序算法把转化率提高了20%多，也让我们看到了技术的价值和追求极致的工作精神。

今天，京东仍然不断地优化着商品评价，增加标签、晒单、问答等功能，创造公平、透明的交易环境，给用户提供最大价值的服务。

对于未来的京东商城来说，除了用户体验还是用户体验，我们追求的是把用户体验做到极致。这样的体验优化，大到一个购物流程，小到用户的一次点击，我们都要让它变得简单而快乐。而这也正是京东人的使命。

京东在不断发展，京东商城网站也在不断进步，不断优化。刚开始只有42个论坛用户，截至2014年第二季度，京东活跃用户量达到3810万，京东网站10年间经过了急速发展的历程。这背后，是京东无数产品经理、工程师、项目经理们的不懈努力，是他们的青春和汗水，伴随着京东，一路向前。（作者/王宇）

# 6 兄弟情——与腾讯ECC 的融合

## 写在融合前

2014年3月10日上午，京东集团与腾讯控股有限公司宣布建立战略合作伙伴关系。京东收购腾讯B2C平台QQ网购和C2C平台拍拍网的100%权益、物流人员和资产，以及易迅网的少数股权，并购买易迅网剩余股权的权利；腾讯以约2.15亿美元收购京东3.5亿多股普通股股份，占上市前在外流通京东普通股的15%；同时，腾讯向京东提供微信和手机QQ客户端的一级入口位置及其他主要平台的支持，以助力京东在实物电商领域的发展，双方还会在在线支付服务方面进行合作，以提升顾客的网购体验。



京东集团与腾讯控股有限公司负责人合影

这次的战略合作在内部被定义为融合，ECC 业务全部交给京东来规划，完成从人员再到业务到系统的全部无缝对接。而作为研发系统的战略项目管理团队，我们无疑承担了系统的融合工作。

在此次战略合作中，双方的合作框架包括：

1. 双方将在电商领域展开战略合作；
2. 腾讯为京东提供移动平台应用的合作支持；
3. 双方将在移动支付上展开合作；
4. 在若干合作领域，京东将成为腾讯的优先合作伙伴；
5. 腾讯将不再从事和京东有竞争的业务。

具体到业务层面，腾讯目前所有的电商业务，如 QQ 网购及拍拍网（包括物流业务）将全部整合进京东，易迅保持独立品牌继续运营，京东将收购易迅网少数的股权，并享有购买易迅网剩余股权的权利。同时，腾讯将在手机 QQ、微信等战略级移动产品上，为京东提供一级入口以及其他关键平台支持，帮助京东发展移动电商业务。

皓瑜在谈及此次合作时指出，它对京东的影响显而易见。首先，以往京东以自营为主的 B2C 业务，将变成自营 B2C+平台 B2C+C2C 的完整电商生态系统。其次，在移动端，腾讯微信的流量，以及三四线顾客群的覆盖，将与京东形成优势互补。

融合之路从这一刻真正开始了，又是一个充满挑战与刺激的旅行。之后，京东的电商业务将会迈向一个更广阔的平台。

## 京东与 ECC 融合之易迅

### 易迅京东店

京东与易迅的业务整合方式是，京东在易迅网上设置“易迅京东店”，顾客可从易迅京东店选择京东商品，订单履约由京东完成。这样可以在现阶段丰富易迅的产品，维持其现有的流量和固定的资源。

2014 年 4 月，双方正式进行物流合作。在易迅下单时，可以看到配送方式改为“易迅京东快递”或“京东第三方配送”。同年 5 月 9 日，京东在易迅网站设置“易迅京东店”，以进一步丰富易迅商品。顾客可首先从易迅京东店选择京东大家电商品，订单履约由京东大家电运营中心完成。随后，日百、3 C 产品也一步步推到易迅京东店。

研发部负责与易迅的技术对接，主要包括商品、订单和库存的对接。简单来说，就是把京东商品的库存共享至易迅，将在易迅产生的订单接过来生产，对接涉及下单、配送、仓储、售后等所有的环节。在团队的努力下，系统对接在 2014 年 5 月 9 日完成最后阶段的上线，整体已近尾声。接下来根据业务的需要，采销人员按节奏把产品一步步推到易迅京东店。

这个过程并不容易，系统融合比想象中要复杂得多。双方的体量都很大，易迅的主流系统有十几个，京东的主流系统更有几十个，而易迅在被腾讯收购之后又对系统进行了几次调整，其复杂性可想而知。技术融合是摆在双方研发团队面前一个困难重重的大课题，怎样最快解决问题、推出商品，对双方都是很大的考验。

## 快速沟通 加速融合

产品经理宝桓回忆起这个过程时谈到，他们把一款商品录入易迅系统，并规划了买一赠一的活动，所有的数据都显示无误，理论上录入之后是不会出问题的。但实际执行的时候，却只能做到单品折价，不得不重新修改。“最初我们总有一些想当然，但实际上事件并没那么简单，每个系统都有自己设计上的优缺点，总有一些细节差异存在”，宝桓说。而解决这些细节问题的方法，只有沟通。

双方的研发体系庞大，人员众多，系统对接工作涉及一百多人，沟通过程中怎样快速找到相关人，信息怎样准确传递，都是最初困扰大家的问题。有时候打电话沟通不清楚，就当面沟通——出差。深圳、上海、北京，很多人成了往返三地之间的“空中飞人”。



ECC 团队 Henry 在做团队介绍



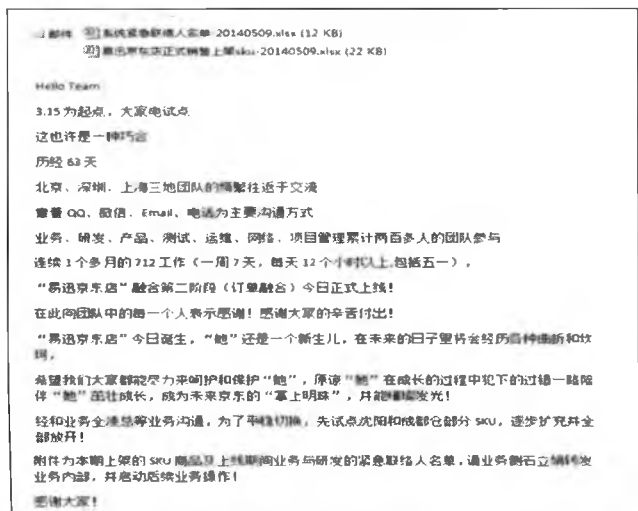
深圳、上海、北京的同学们在北京讨论方案



## 做中间转换接口匹配数据

而让何方印象深刻的是，他和几个研发同事去深圳确认一些方案细节，当时京东、腾讯电商、易迅三方在一起共同提出了三四个方案，这个过程非常艰辛。到最终拍板的时候，为了抢时间和提效率，他们没有按照既定的请示流程，而是直接在会上一个个碰撞、沟通，高效地把方案确定下来，再整体提交领导。“现在回头想想，我们都有点傻大胆，自己做了很多决定。但是那个时候，为了抢时间，一定要有人站出来承担责任”，何方说。

在这次融合启动前，我们特意安排大数据部门抽取了双方系统最近一年的历史订单，进行对比分析。团队拿到这组数据后，和研发、业务的老大们多次沟通和分析（当中也补充了多组支撑数据）后发现，两个网站的用户群体的偏好确实存在很大的差异，双方网站用户的重合度非常低。这就对我们的融合提出了更高的要求。用户和流量是电商平台最重要的资产，我们的所有产品对接和系统设计，都要以用户体验为第一，不能改变用户体验。否则，可能会引起用户的流失，进而导致流量的下降。但双方系统都各自运行了很多年，不管是用户体验，还是业务场景上都有很大的不同，比如在配送环节，易迅有一日三送，而京东是定时达；在仓库生产环节，一边是三级地址，一边是四级地址，等等。这些都不能简单统一为一个，而必须能互相兼容，且用户体验只能增强。类似这样的不同点多达几十处，最后的方案就是做一个中间转换接口，分别来解析和匹配两边的数据对接，并通过历史记忆来自动匹配。



易迅京东店项目正式上线公告

## 与易迅深度合作：强化 B2C 领域

易迅网作为独立品牌继续运营，将充分结合京东的领先优势，继续探索平台的运营模式。此外，双方将以提升客户体验为共同目标，互取所长，在平台建设、产品营销，以及客户及地区拓展上展开协作。

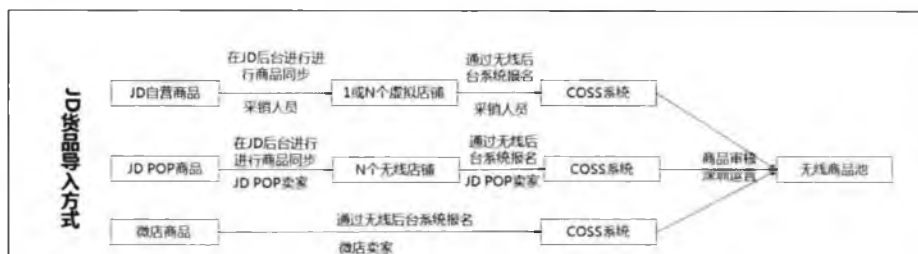
# 京东与 ECC 融合之 QQ 网购

## 最初的握手

2014 年 3 月 10 日京东和腾讯战略协议刚刚签署后的第一周，营销研发部的马总就带领一行人等来到深圳，对京东以后在微信端和手 Q 端如何销售京东自营和 POP 商品，进行了第一轮技术方案讨论和对接。此行敲定了一个大的决策，就是通过系统对接的方式将京东商品通过 ECC 系统，在微店和手 Q 端销售。大方针确定之后，刻不容缓，北京和深圳的产品及技术人员即刻开始了具体实施方案的讨论和对接，融合大幕就此拉开。

## 融合伊始

2014 年 3 月下旬，ECC 和京东融合大战略号角已经吹响，京东平台和 QQ 网购的融合以燎原之势迅速开始落地。在经过几轮北京和深圳的方案讨论后，快速确定了融合目标：首先实现京东 POP 商品同步至微信端，所有交易等相关后台系统使用京东现有后台，原有网购 COSS 系统实现商品在微信前端的发布与展示，实现时间为 2014 年 4 月 30 日。也就是在短短一个多月的时间内，要实现商品展示、交易流程、促销逻辑、售后等系统和原有 QQ 网购系统全面打通和对齐，完成在微信场景内的一站式闭环购物体验。在此基础上，再实现自营商品发布微店，以及和主站对齐等其他功能。



京东商品导入无线端流程

不容许任何的迟疑，北京即刻成立了专门的项目小组，并在京东会议室资源非常紧张的情况下申请了一间专用会议室作为战时指挥部，主要项目组成员随即打点行囊入驻、封闭，至此微店、手Q、微信融合项目轰轰烈烈地宣告启动。



当时的战时作战部

由于最初需求派下来的时候，时间要求严格，领导对于需求的描述就概况为一句话“可以实现在微信上卖京东的商品”。在这一句看似简单的需求背后，融合的确需要北京、深圳两地多个业务、产品、研发团队无数次的沟通、修正和冥思苦想。按照最初的方案，微信给京东购物开放两个入口，二级入口为“我的银行卡”→“精选商品”购物入口，一级入口开放为“新发现”→“购物”入口。在进入京东购物首页后，设置3个栏目，分别为“新发现”、“品牌”和“聚惠”。3个栏目分别定为：“新发现”针对首发、预售、潮流商品；“品牌”主要接入厂家品牌特卖；“聚惠”则为特价普通商品，如团购等。



### 聚惠、品牌特卖、新发现

此次融合，对于京东而言，是空前的。不同的文化、不同的地域、独立的系统，要全面打通面临着各种技术、商务、工作方式等方面的不同。对于项目组而言，每天面临的各种问题从四面八方并行而来，商家、商品、订单、支付、售后等各个环节随时会有新问题、暴露新风险。需求上的、技术上的以及要平衡各个相关方的利益而产生的各种待确认问题，像一场又一场密集的流星雨，每天在项目指挥部的上空火花带闪电般接踵而至。好在大家配合默契，彪哥和彩宝负责总体对接深圳需求，接口和网络互连问题交给了彪哥，飞哥负责跟进 POP 商家及商品的开发。面对每天紧张繁重并且高并发的的工作，合理的工作划分和拆解显得尤为重要。然而，就在 2014 年 4 月初 POP 商家开微店的需求刚刚提出不久，融合亢奋之情未艾之际，深圳团队也就是原 QQ 网购的同事，派出超强代表团来到北京，开始了第二轮大范围交流。



深圳团队

此行深圳团队又提出了新的目标。在实现 POP 商家在微信端开微店的同时，后续马上要引入京东搜索，并且要实现京东自营商品在微信端的发布，最重要的一点是，618 是京东店庆的日子，整个 6 月为京东的店庆月，京东从 2014 年 6 月 1 日开始做活动，所有相关需求要赶在 2014 年 5 月底之前全面上线，才能保证一级入口的对外开放，从而赶上 2014 年 6 月的店庆大促，如果错失了这次冲刺的时机，就只能等到双 11 了。而此次一级入口的位置就在“游戏中心”的下面，618 还要配合小红点推出，真是一波还未平息，一波又来侵袭，时不我待，唯有受命！在会上，大家初步拟定这个版本的实现日期为 2014 年 5 月 25 日，而当天是 2014 年 4 月 16 日。



京东自营商品微店

## 项目进行时

每天早晨 10 点 10 分，“阿姆斯特丹”的电话铃声都会准时响起。深圳团队同事的电话又如约而至，这是 QQ 网购融合项目组北京团队和深圳团队每天雷打不动的晨会时间，每到这个时候双方都要对齐一下当前各自的问题，凡是涉及的主要人员此时都要出现在双方各自的会场。那个阶段，最常见的场景就是在阿姆斯特丹会议室里站满了人。晨会结束后，如果你以为能够喘口气，以放松的心情眺望一下窗外不远处的鸟巢，陶冶一下情操再开始一天的工作，那你就错了。因为你根本就没有站起来抒情的时间。刚刚放下晨会的电话，几个人的电话就轮番响起，待确认需求，待解决问题，各种突发事件……从 2014 年 4 月初到 5 月最后上线甚至直至 618，每天都是这种状态，需求像雪片般飘来，讨论会一个接着

一个，解决了一个问题又冒出 N 个问题。你必须学会同时处理多个紧急且重要的问题，比如一边接着一个问题电话一边用邮件推动另一个问题的进展。那些日子作为加班的我有时看着研发的同事们，经常怀疑他们能不能撑得住，尤其是交易组的同学，作为整个交易流程中最基础一环，经常为了一次上线都要持续工作到凌晨。上天不会辜负每一个持之以恒的人，大家的坚持终于有了回报，2014 年 4 月中旬我们撑到了首个功能 POP 商家系统的成功上线。接下来，各个系统便开始陆续上线。与此同时，新的需求也不断被提出，每一周都有新的迭代版本发布。用户体验的同事经常兴奋地跑过来对我说，今天又看到新功能了！当然，在为项目的每一个阶段性成果感到开心的同时，日程也一天比一天紧张。

忙碌的日子总是过得飞快，转眼间 2014 年 4 月 30 日微店、手 Q、微信融合项目迎来了第一个重要里程碑——第一阶段的上线！得益于之前大家夜以继日的奋斗，在这个关键节点，过程虽然严肃紧张，但一天下来，由于北京、深圳、宿迁三地的同事积极配合，协调有序，各种验证过程都很顺利。本来大家都已做好了通宵奋战的准备，晚上 11 点多，北京测试接口人张琪突然来到会议室，满脸笑意地宣布，今日所有验证均已成功通过。幸福来得有点突然，一个多月以来的焦虑和疲惫在那一瞬间终于可以释放了，今天可以早点回家睡觉了。接下来，也能够怀着忐忑的心情休个小长假了。2014 年 5 月 4 日，我们迎来了微店第一单！接下来，又陆陆续续地实现了将京东自营商品发布到微店、货到付款、大家电配送预约等各项功能。等到 2014 年 5 月 25 日到来之时，虽然还存在这样那样的小问题，但是之前计划的两个大版本的所有需求均已全部实现。

2014 年 5 月 26 日，微信大入口成功上线，那一刻感觉，付出的所有汗水都值得了！

---

4月	• 5月4日	• 5月26日微信大	• 6月4日微信大	• 6月18日手Q二	• 8月4日手Q一
• 人工对接盘货	pop商品发布微	入口灰度发布	入口全量发布	级入口全量发	级入口全量发
• 运营、卖家报名	店			布	布
系统培训	• 5月8日				
	自营商品发布微				
	店				

QQ网购融合时间轴

## 写在项目结尾

还记得项目伊始，当初 QQ 网购的负责人 Fiona 曾经说过，之前微信“发现”下的游戏入口上线用了 9 个月时间。而此次京东和 QQ 网购的融合，到最后实现“购物”一级入口的上线，只用了短短两个多月的时间。想到这里，大多数人应该和我一样自豪。项目的成功离不开优秀的团队，在整个项目进行的过程中，虽

然北京和深圳的团队远隔千里，但是配合得非常默契。为了共同的使命和愿景，大家在融合的过程中求同存异、互相尊重、忘我工作，项目才得以顺利完成。在此，特别感谢深圳团队的 Allen、Double、Raywu 和晓佑。

这次微店、手 Q、微信项目只是融合的开始，要实现真正的融合还需要一个漫长的过程，接下来还有很多工作要做，未来也充满了变数，但是在这个历史性的节点，能够亲历京东这次变革带来的欣喜和痛楚，也是我们每个人成长的重要部分。

## 京东与 ECC 融合之拍拍

### 拍拍兴起人与成长

时间退回到 2003 年的春天，国内的传统行业波澜不惊，零售业的半壁江山由继承者们主宰，人们一如既往地超市和专卖店购物结账。谁都不会料到，命运洗牌的机会来得如此突然。一场非典席卷过后，互联网电商凭借敏锐的嗅觉和便捷的服务，强势登陆千家万户，网购带来的新潮体验立刻在小伙伴儿们之间口口相传。淘宝网、京东多媒体等电商公司应运而生，准备赶上好时运大展身手，即时通讯霸主腾讯内部也在酝酿一盘很大的棋局。

转眼到了 2005 年 9 月 12 日，腾讯悄然发布 C2C 网站“拍拍网”，在互联网风头正盛的时候，加入了这场看不见硝烟的战争。从此，江湖上多了拍拍的故事：运营一白天就跻身“全球网站流量排名”前 500 强（来自 Alexa 数据），创下电子商务网站进入全球网站 500 强最快的纪录，业界首创“边聊边买”、“买家与卖家信用分离制度”等专利；通过短短一年的运营，拍拍网已经与易趣、淘宝共同成为中国最有影响力的 3 大 C2C 平台；2007 年跃居国内 C2C 网站第二。



改版过程中的拍拍首页

## 拍拍重塑

在这条且行且创新的路路上，电商领域最凶猛的投资如期而至。2014年3月10日，京东与腾讯宣布强强联合，QQ网购、拍拍的电商和物流部门并入京东。在原有B2C的基础上，京东开始涉足C2C领域，进一步丰富原有的电商生态，通过引入海量中小商家和长尾商品完成对消费者需求的全面满足。当今时代，得用户者得天下。作为一名活跃的普通用户，咱也想代表自己体验一下重塑后的拍拍是否和京东平台一样简单便捷。

转眼的功夫，与拍拍的缘分不期而至。2014年5月5日，领导指派我作为项目经理推动京东拍拍账户打通项目。不难理解，登录打通是第一步，后续的所有运营都需要其支撑，项目的重要性可见一斑。我们要做的，就是与ECC同事合作，将京东账户体系引入拍拍，实现拍拍与京东用户注册及登录的整体融合。



风格明朗的拍拍新首页



面对期待，项目组士气大涨，同时也经受着各种考验：团队分布在北京和深圳两地，京东和拍拍是完全不同的两套系统和设计理念。系统对接，人员对接，一时间觉得自己肩上扛着无比重要的责任。要知道，无论是实现商品 SKU 的批量互通，还是虚拟资产的统一管理，没有账户打通先行，没有系统无缝连接作为基础，这些完善的功能和完美的体验将成为纸上谈兵。时间紧，任务重，深圳同事即刻动身，周末便马不停蹄地赶到了北京。次日一早，忘记了旅途劳顿，项目组开始对接产品和设计方案，团队在紧张而有序的氛围中高效行进着。有过面红耳赤的争论，有过推翻再来的否定，有过悬而未决的纠结，最终方案向领导汇报后，择优录用，达成一致。仅仅 1 周时间，项目的产品、研发、测试及整体计划就敲定了。



京东账户登录拍拍的场景设计

这个过程着实让人留恋和难忘。详细设计稿、具体实现流程、技术方案以及安全策略，每项内容都经过项目组紧密沟通，第一时间进行信息共享，大家团结一致，呵护着这个极其重要又尚在襁褓中的婴儿。为了同一个目标，我们推翻了 N 多种方案，否决了 N 多条想法，过滤了 N 多种场景，坚定地落实着自己的职责，同时也在挑战着自己的神经极限。产品人员的实时对接，开发人员的加班赶工，测试人员的周末加班，项目组的无数条群组讨论，都见证着我们的汗水和努力，都表达着我们的决心和执行力。在项目后期还发生了一个小插曲：京东已经发布上线，恰逢拍拍的结算第三方财付通平台进行容灾演习，当时项目组陷入了两难境地。如果不等待财付通的更新发布，直接上线账户打通的改造，那么可以如期完成项目，但是原有接口不支持用京东账号登录拍拍的用户下单付费；如果等待财付通演习完毕，又会导致项目比原定计划延期上线。这可是公司战略级任务，按理说一刻都容不得耽搁。抓耳挠腮地犹豫一阵子，决定跟领导汇报并听取

相应安排。比较踏实的是，领导最终指示项目组不能牺牲用户体验，通过延迟发布时间来解决这个问题，平稳地将上线风险控制到最低。当凌晨结束上线的时候，项目组如释重负。线上验证顺利通过，两地同事来了个隔空聚会。



一家亲：北京拍拍融合团队+深圳拍拍融合团队隔空庆祝合影

当我再次坐在窗明几净的工位，享受着美好阳光，输入拍拍网址（<http://www.paipai.com/>）进行浏览时，看到自己负责的项目正在线上欢快地运行，油然而生的骄傲是言语无法表达的。当然，上线后对数据和运行进行监控时，也会觉得分外亲切。不管是使用京东账号直接登录拍拍，还是在拍拍网注册京东账号，一切流程超级顺畅，So easy！



上线后的拍拍登录页面

在这个再熟悉不过的页面，输入自己的京东账号信息，尝试着登录→选择好心仪的商品→加入购物车→填写收货地址→提交订单，一路畅通无阻，这不就是

用户想要的吗？

一路走来，团队成员收货了很多感动、很多感恩、很多感谢。在项目实施过程中，无论是一直支持和关注的公司、部门领导，还是不离不弃并肩作战的两地同学，是  $1+1>2$  的公式让我们实现了公司的目标，也展现了自己的价值。黄轩，是拍拍的后台开发，通过和他沟通，或许我们可以看到拍拍一路走来的真实缩影：

我来腾讯 5 年了。刚开始一年是在 QQ 会员，后面随团队一起转战 QQ 商城、QQ 网购、易迅，现在又接手拍拍，负责后台开发。互联网是前沿技术跟传统业务结合的阵地，能尝试许多新鲜的东西，所以我一直留在互联网。接手过很多业务，每一个都需要在很短时间熟悉业务、架构，然后上阵冲锋，现在已经驾轻就熟。

做电商最难忘的经历必然是大促，特别是看到努力优化的系统能够运行良好。易迅的网站系统最初抗压能力一般，流量稍高时就有可能出现打不开的情况，我们改造了三四个月，到了年底大促各方反馈比去年平稳许多，大家很有成就感。

京东的优势是商业思维。之前腾讯电商可能是抱着做互联网产品的心态和经验来做电商，做出来的东西没有碰触到传统商业的痛点。京东透彻的商业认识，加上腾讯产品技术的辅助，很有希望产生新的效果。比如说拍拍，京东接手后的战略让我看到了不同于以往的思路，期望能挖掘拍拍这个老品牌的潜力，发扬光大。

## 拍拍梦想

新的拍拍网，重燃我们的希望。2014 年 5 月中旬，京东的商品已经可以实现批量导入拍拍。接下来，拍拍的二手拍卖业务与京东备件库商品完成对接，进一步丰富和补充了新拍拍网的二手频道，除了在规则和体验上的重塑，新拍拍网将在业务模式上进行创新，二手拍卖市场、京东传统的 3C、母婴、汽车等商品类别，以及服装、海外购、农业产品等，都将是拍拍重点发力建设的品类。未来，拍拍将会被打造成一个公正、诚信、可信赖的电商平台，在移动端发力，凭借新技术和京东、腾讯的支持，重构流量分配原则，推动移动电商产业的发展，为卖家和整个生态链创造新的价值。

目前的电商发展势头正盛，未来的电商之路扑朔迷离，格局瞬息万变也暗流涌动，谁去谁留，花落谁家都是未知数。作为一个伴随电商共成长的同事，借此也祝愿拍拍在新的起点，与公司一起扬帆起航，实现公司的价值，社会的价值，这也是我们京东人的价值！

## 写在融合后：J+T=今天

通过这次融合，京东在朝着构建完整的电商生态圈的道路上又前进了一大步。而腾讯从线上到线下，真正实现一个大生活圈闭环——构建新的移动电商生态圈。双方将携手向中国互联网和移动互联网用户提供更卓越的电子商务服务。

在双方战略合作的绚烂蓝图背后，有几千名 ECC 同胞，他们面临选择，充满了对过去努力奋斗的不舍。他们更憧憬未来，期待着在新的平台上实现更大的梦想！加入京东，是一场新开始。

他们大声说：你好，我是京东人，我来自腾讯。（作者/章爱文、暴丹、王晓琼、李昂）

# 7

## 猩球崛起——POP传奇

### POP 平台简介

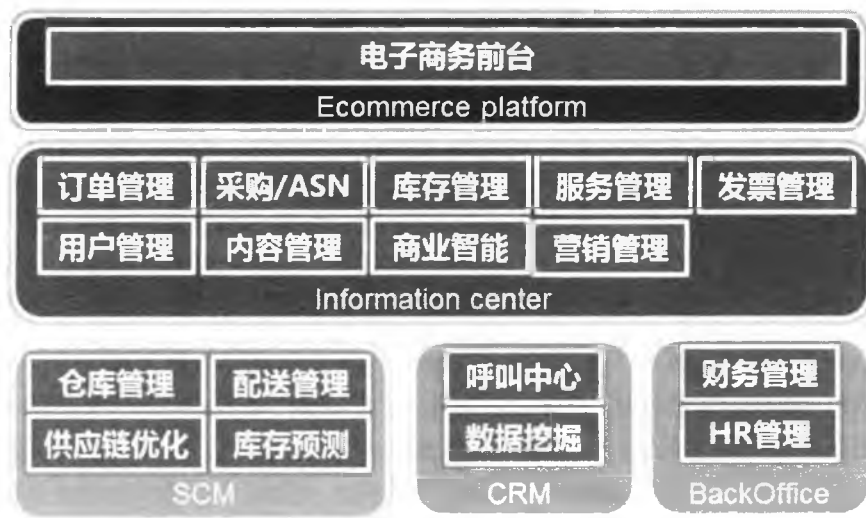
纵观各种互联网应用的发展趋势，最后的王者之争实质上是开放平台之争。行业领跑者通过自身的强大流量资源和基础设施，扶持竞争优势不明显的中小型企业成长，最终成为行业孵化器，制定行业通行标准。而 POP 正是京东的开放平台，也是京东大平台化战略的基石。

在京东决定做 POP 之前，没有人知道京东的开放平台应该是什么样的，但是京东很清楚平台不仅要追求量，更要保证商家的资质和商品的品质，因为它是京东开放平台的根基。面对一个新的业务，京东人再次拿出了当初创业时的态度和决心，将自己定位成一个创业团队。而 POP 就是京东的另一次创业也可能是迄今为止，京东最大的一个创业项目！

2010 年 3 月，京东 POP 项目正式启动，进入紧张的产品规划阶段。由于时间紧迫，特申请了苏州街银丰大厦二层的 202 会议室进行半封闭开发，团队每天起早贪黑、加班加点，关键节点以及关键问题大家一起头脑风暴。马总每天和大家一起办公，他的办公桌就设在会议室的一个角落。



业务模式



POP 架构

根据国内的现状，在模式上做很大的调整和改进，设计出很多种商家的合作模式，充分发挥京东自身的仓储物流优势，只要商家有资源，资质达标，都可以入驻京东开放平台，解决商家的后顾之忧。



POP 交易流程

1. **FBP (Fulfillment By POP) 模式**: 商家采用服务①②③④⑤⑥，商品入京东仓储，用京东物流，支持自提、货到付款，开京东发票；
2. **LBP (Logistics By POP) 模式**: 商家采用服务①②④⑤⑥，不使用仓储服务，订单产生后送往京东配送中心，支持自提，货到付款，开京东发票；
3. **SOPL (Sale On POP Logistics) 模式**: 商家采用服务①②④⑤⑥，不使用仓储服务，订单产生后送往京东配送中心，支持自提，货到付款，商家负责开具发票；
4. **SOP (Sale On POP) 模式**: 商家采用服务①②，使用三方物流，商家负责开具发票；
5. **LBV (Logistics By Vendor) 模式**: 商家采用服务①②，使用三方物流，京东负责开具发票。

POP 合作模式的发明使老马的团队有了“发明家”的美名。为了打通各种模式，团队产品、研发成员深入到郊区的仓储、分拣中心、配送一线，在一线和兄弟部门一起探讨、沟通，将这些模式变得可实现、业务层面可操作，且不影响京东自营业务。

作为创业团队，POP 团队一直以来都十分稳定和团结。整个团队干劲十足，经常在工作过程中因为一个问题进行激烈争论，从而得出很多优秀的思路。每天晚上产品经理都会将自己设计的原型拿出来，让大家提意见，然后进行整理修改。在大家的努力下，系统的功能框架基本搭建完成。



POP 系统功能框架

随着业务模式和整体框架的确定，POP 频道、商家管理、店铺管理、商品管理、订单管理、仓储管理、物流管理、客服/售后管理、营销管理、报表统计等大的功能模块也确定了下来。模块的规划和流程的树立就需要产品经理来做了。

经内部分工后，产品经理各司其职。他们每天和财务、法务部门进行沟通，并一次次地修改商家入驻的流程、规则及合同；进行包括类目、商品属性、销售属性、商品编码、图片、SEO 的优化、搜索等在内的商品管理工作；深入仓储物流一线，了解 WMS 系统、物流配送系统，保证平台能平滑地接入现有 ERP 系统；和售后客服人员交流，对平台的逆向流程进行改造；在 ERP 系统财务业务人员和信息系统的同事的共同努力下，打造全新的 POP 结算系统。

每天产品经理白天开会、去业务现场、熟悉业务、了解细节逻辑，晚上整理白天的收获，通过流程图（UML）梳理出业务逻辑，翻译成技术语言。进而将各功能模块通过 Axure 转化为业务部门可操作的系统原型。对于原型，先内部初评，内部发现问题后不断地修改，通过后，业务部门再进行评审，根据业务部门提出的意见和需求进行一次全面修正。

每天下班后，产品经理们都要把“作业”交给老马，老马会在产品经理部在场的情况下对原型进行一一评审，并提出问题，帮助产品经理进行不断的优化。

经过一轮又一轮的评审、修改，分模块原型终于出来了，接下来的工作是将这些模块整合起来，包括各模块在功能上、逻辑上的衔接及相互依赖管理。整体的商家管理后台、运营管理后台第一稿出来了后，产品经理需要将这些原型、逻辑、业务场景、字段信息通过 PRD 表述清晰。





POP 创业团队产品规划团队合影

2010 年 5 月，POP 平台原型、PRD 陆续交付研发，项目进入了紧张的开发阶段。在整个研发过程中开发人员都和公司各部门就业务和逻辑问题进行不断沟通。

## POP 平台搭建

与产品规划同步进行的是研发团队的组建，京东当时的技术架构还是基于 .Net 的，研发部负责人老李已经预见到未来必定要迁移到 Java 架构，因此，POP 项目决定采用 Java 架构，但当时都是 .Net 的开发人员，急需熟悉 Java 技术的人才。这时候思勇加入，成为京东 Java 技术开创人。虽然当时仅有老杨和杨凯两名正式 Java 开发人员，但 POP 研发团队还是于 2009 年 3 月开始了筹建工作。正在老马和老杨为招聘进度缓慢发愁时，有一支 Java 开发团队恰逢其时地加入了京东。

### 千寻团队的加入

经过 2009 年的快速增长，京东开始意识到扩充品类的必要性。在非标品中，以服装类为代表的时尚电商是当下最热门的品类，但是如何进入这个领域呢？当初还是 SK 集团子公司的千寻网，就逐步进入了老刘的视野。当时的千寻网在时

尚电商领域做得还不错，不仅有时尚品类的采销人才和经验，技术团队也很优秀。王彪就是那时候千寻技术团队的负责人。

2010 年 3 月底，王彪带领李文利、高飞、孙炳蔚、张峰、韩世超千寻研发核心加入京东，组成了 POP 平台创业时期的核心成员组。王彪记得刚过来就有两位京东的同事过来找他们，说之前听过千寻的技术团队，想让他们指导学习 Java 方面的知识。其中一位同事就是薛韬，他也是 POP 研发初始团队成员之一，后转战 POP 平台、团购、虚拟产品等多条产品线，立下了赫赫战功。

2010 年 4 月初，开始讨论技术方案，原来的千寻网就是平台模式，所有的商品都是韩国那边的商家通过接口同步到网站后，进行翻译、编辑后再上架。产生的订单也通过接口同步给韩国的商家，商家订单生产后再将信息回传。这与 POP 平台的模式非常相似，因此，POP 平台的数据表设计和流程梳理都比较顺畅。

### 研发故事节选：第一个 POP 商品

商家和商品最先确定了需求，因此启动早一点，2010 年 4 月开始设计表结构，2010 年 5 月初正式进入开发。最初的设计是将 POP 商品作为主数据，原来自营的商品同步过来，但随着项目的进行，发现要将自营商品同步过来，系统修改度太大。当时研发的主要力量在进行 618 店庆备战，根本抽不出人力进行支持，最后只好改变方案，将 POP 商品同步到自营系统。大家知道，商品数据最复杂的就 是类目属性，类目属性的映射和同步是最难实现的环节，而且商品数据结构非常复杂，涉及十几张数据表，每个字段都需要进行映射。这是个艰巨的任务，由抽调来张峰负责搞定。经过大家历时 3 个月的共同努力，第一个 POP 商品终于上线了。



第一个 POP 商品



网友热评

以下为开发人员高飞的回忆：

当时我还是刚刚进入电商互联网的 23 岁的小屁孩，充满着对互联网技术的热爱，跟随我之前的研发老大王彪于 2010 年 3 月从千寻网来到了京东。那时京东办公区还在苏州街银丰大厦，我很快在这里认识到一批非常优秀且经验丰富的研发人员，大家都是从各大互联网公司怀揣着梦想来到这里。

公司 2010 年年初开始启动 POP 开放平台项目。商品是所有业务的起源，理所当然成为了最重要最紧急的技术屏障：从标类到非标类，从固定到灵活，从现有商品系统到支持 POP 商品。商品系统最主要最复杂的部分是商品分类、属性，自营商品系统看上去无法兼容 POP 商品。老杨带领着杨凯、张峰讨论商品分类、商品属性、商品发布、商品到前台展示的问题，通过一轮轮的头脑风暴，来抉择我们应该打造一套全新的商品系统，还是在现有商品体系上进行改造来兼容自营和 POP 的商品。最终我们向老李寻求了帮助，老李考虑到项目研发的周期选择了后者。现在看来，在当时的情况下这样的选择是完全正确的，从前台展示商品到仓储配送等购物体系中的所有系统都和商品的设计方案息息相关，再愚而未决只会浪费原本就很紧张的项目时间。

方向指明后很快我们制定了方案：分类属性数据使用 JD 自营系统、销售属性数据使用 POP 系统，使用数据同步将 POP 商品数据从 Oracle 同步到自营商品数据库（SQL Server），Sku、Ware 号段从 10 亿开始。这个数据中间件还是京东当时最主流开发语言 .Net 开发完成的。经历过无数次加班磨合，平日里仰慕的技术牛人也逐步成为了好朋友。2010 年 7 月商品系统研发完成进行灰度测试，从创建商家开始到发布商品操作页面每一步操作都是在检验我们的成果，经过一次次

的报错和调试，最终杨凯成功地在 POP 前台展示了第一个商品。

这个 SKU 正是我们第一个 POP SKU，当时杨凯只是抱着试一试的心态使用这款 Nokia 2100 进行灰度测试，谁知道居然发布成功，种种原因促使这款 SKU 逆天地上上了京东首页，引起用户一阵围观。



高飞在给运营同事培训

## POP 平台亮相

2010 年 9 月 15 日，经过产品和研发人员 4 个多月的半封闭开发和努力，以及各业务部门的通力合作，POP 平台正式进入试运营阶段。次日，刘总在派代网年会上正式对外宣布了这一消息。

在平台进入研发的同时，POP 团队进行了商家及运营人员的招募。在大家努力下，在系统上线试运营时，合作商家已经有了几十家。

2010 年 11 月，POP 平台专属频道“品牌直销”上线。



品牌直销频道上线

## POP 平台演进历程

自 2010 年 9 月正式上线之后，POP 开放平台的发展就驶入了快车道，业务以每年百分之几百的速度急剧增长，系统的压力和挑战也自然成几何系数增长，在应对业务压力的过程中，POP 平台的技术架构和业务系统随需而动，不断地蜕变和进化，POP 研发的兄弟们也在这种挑战中不断超越和成长，痛并快乐着。

### 订单系统升级

订单机构改造（2010 年 11 月）。系统刚上线一个多月，原来订单系统设计上的一个缺陷就暴露了出来：订单上的商家 ID 用了原来京东的订单机构字段，因为京东的业务模式为各大区是不同的子公司，财务记账及结算都需要落实到各个公司上面，也就是机构；FBP 和 LBP 类型的订单是代销模式，需要京东开发票，因此就需要落实到一个具体的机构；SOP 为商家开发票时涉及佣金扣点，因此也需要确定一个机构进行结算。因此，订单机构是不能用来记录商家 ID 的，当时

的订单表已经非常复杂，不能再增加字段，所以就增加了一个订单扩展表。

这是 POP 上线以来面临的第一次系统升级，这个项目非常紧张，记忆最深刻的是上线那天，大概到晚上 3 点才完成上线和数据迁移和清洗。

订单离线生产（2011 年 3 月开始）。POP 业务发展非常迅猛，经过 3 个多月的发展，POP 订单每天就可达到了 10000 单左右，原来商家所有的查看及订单操作都是实时调用订单中间件，当时订单库还是用的 SQL Server 主从复制，数据库复制已经不堪重负，主从复制经常出现延迟，一到高峰期经常演示两三个小时，所以每到下午 3-5 点商家订单出库高峰期，商家就看不到新订单，产品人员没有办法，最后只得建议商家 8 点后再操作，想想都汗颜。数据库复制已经成了难以攻克的难题，说实话当时京东也已经将 Windows+SQL Server 的性能发挥到了极致。这个方向基本上无法突破，但问题已经到了非解决不可的地步。

解决之道只能是绕过订单主库和订单中间件，新订单产生后直接发一个消息，POP 系统接收到消息后，将订单信息解析并保存到自己的数据库，这样数据就不会出现延迟了，但如果只是这样做的话，过几年 POP 订单量大了还是会出现无法扩展的悲剧。

为了将订单生产彻底与订单表分离，提出了离线生产的概念，所有的订单生产环节都只针对任务单，任务单就像流水线的一个工序，每个处理环节完成后生产下一环节的任务单，任务单处理完成后即可定时归档，这样保证了任务表的数据量保持最小，性能当然也是最好的。同时，最终的订单状态还是要回传到主数据的，订单状态同步也是采用异步的任务模式。

方向确定后，马上由李文利牵头推进订单离线生产项目，经过一个多月的奋战，新的订单生产系统终于顺利上线。离线生产的上线，解决了订单延时及高峰期间商家无法操作的尴尬状况，为后面的订单量继续增长做好了准备。

小插曲：POP 研发过程中最激烈的争吵发生在订单离线生产项目确定解决方案时。彪哥希望尽快推进，对李文利的方案不是太满意，甚至拍了桌子。不过经过争辩，最后还是采用了李文利的设计方案。这样的情景，在 POP 创业项目推进中是非常常见的，别看为了方案大家争得面红耳赤，但也正因为这样的碰撞，才使整个项目以最优形态面市。当我们多年后，回忆起这些时，感慨良多，这就是创业的激情。

## 系统拆分与服务化

随着商家数量和订单数量的急剧增长，商家对订单的查询量也越来越大，同时，订单 API 的调用量增长也更为迅速。解决海量调用问题的唯一途径就是不走

数据库，而是走缓存，采用 MemCached 或 Redis。但商家查询订单的使用场景是动态条件的搜索查询，代替数据库复杂查询条件的最好办法就是用搜索引擎，Solr 是一个不错的选择，同时，为了实现 SOA 化我们决定建立订单中心。用什么服务调用框架又成了一个难题，那时京东内部的服务框架五花八门，有最原始的 Web Service，也有后来的 Ice，经过综合考虑，采用了轻量级的 Hession 作为服务调用框架由 2011 年 7 月就让刚入职不久的李扬帆同学负责。开始订单中心建设的技术选型预研。

POP 系统大改。系统垂直拆分势在必行，最初的 POP 系统为了快速搭建，只有两个系统：POP-Vender 和 POP-Man。POP-Vender 面向 POP 商家，POP-Man 面向 POP 运营人员。POP-Vender 就是 Shop 端，所有商家后台都在这个工程里，包括商家管理、商品管理、促销管理、订单、配送、仓储、售后、BI 报表等多个模块。

随着开发人员的增加，系统功能的完善，系统越来越复杂，工程的代码量也越来越大，每次发包测试环境，编译就需要 30 分钟。

另外，2011 年年底的一次事故也加速了这一问题的暴露。周末的一天，李总打电话给王彪，说商家后台崩溃了，订单不能出库发货。开始可以通过重启缓解，可每次重启后不久就会再次崩溃。当时整个商家后台只有两台服务器，但也不至于挂掉。调取 Apache 访问 Log 进行分析后，发现商品操作的 URL 访问频繁，堵塞了其他模块的访问。同一个应用，不同模块之前相互影响，牵一发而动全身，要找到问题的根源往往需要很长时间。

解决这种困境的最好办法就是对系统进行拆分，将不同的模块拆分为子系统可是个大工程，没有两三个月是搞不定的，总不能每天这么煎熬下去。彪哥马上打电话给当时几位核心技术人员：李文利、杨凯、高飞、管亮祺，确定了下面几项任务：

主题： 紧急：shop端崩溃 发货阻塞

李总，

昨天杨凯、高飞来公司，12点多处理好了。

今天我们会紧急突击以下几项工作，争取彻底解决问题。

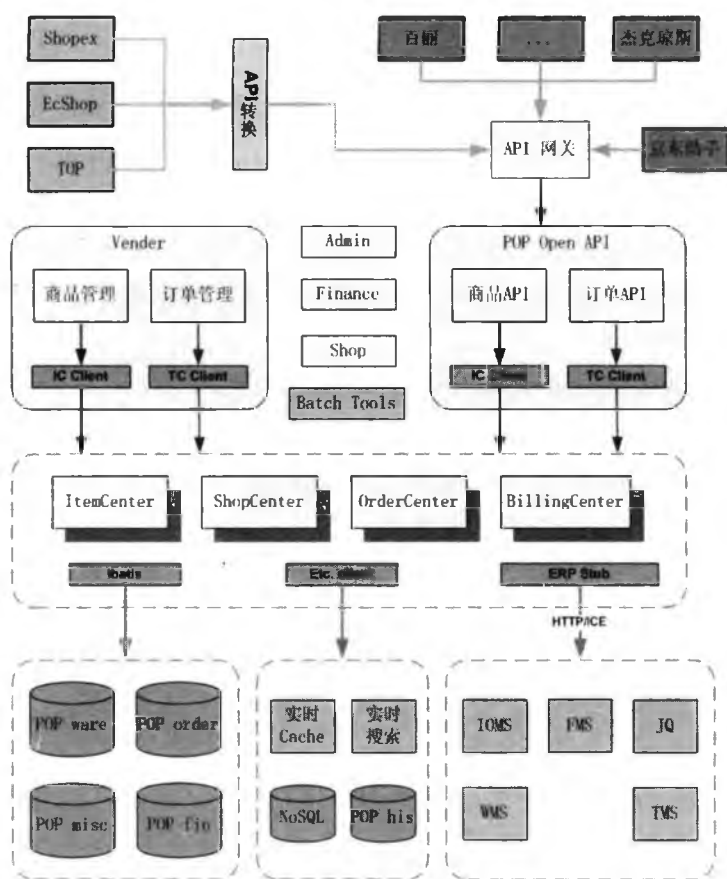
- 1、 分析 vender 访问日志，将访问量上和响应时间长的页面统计出来，重点改进性能；
- 2、 将调用外部接口的超时时间设短，防止因外部接口慢造成堵塞；
- 3、 将重要功能模块（商品、订单、促销）分离部署，避免相互影响。

核心技术人员处理的几项紧急任务

代码层面先不拆分，只是分开部署，将核心功能拆部署 ware.shop.360buy.com、order.shop.360buy.com、fin.shop.360buy.com，同时申请增加了 6 台服务器，分别用于单独部署这 3 个核心模块。

兄弟们一直战斗到当天晚上 11 点半，终于完成了系统的拆部署。果然，拆分之后订单的功能正常了，速度恢复正常。

这么做并没有真正实现系统拆分，于是项目组马上启动了系统拆分与 SOA 化，将服务中心与应用分开，形成如下所示的系统架构。



升级后的 POP 系统架构

## 在线招商平台建设

在 POP 平台建立起来的第一年，商家管理方面仍然停留在基本信息管理的层面。商家的各种信息、经营资质、合同、协议依然通过线下的方式进行管理，



导致业务人员工作量大、交互周期很长、纸质信息管理检索困难、整体商家管理效率低下等很多问题。另一方面，大量同类平台商家转战京东平台，经过横向对比，他们从入驻京东的第一步就开始抱怨：

- 还需要签订纸质合同？
- 入驻周期要一个多月？
- 品牌资质也需要通过线下方式邮寄？
- 品牌还需要运营手工添加？
- 二级域名服务都没有提供！
- .....

类似的抱怨还有很多，这些声音让我们真正感受到还有许许多多的工作要做……

在那个炎热的夏天，业务部门 VP、总监、研发总监、经理、产品经理、业务人员、法务、财务税务等分模块分部门分主题展开多次讨论，挖掘京东平台商家管理的业务特色，并结合与其他平台的竞品分析，我们重新定义了商家管理，确定了以下发展思路：

1. 业务规则系统化，即将平台商业规则真正融合入 IT 系统中，复用相对完善的实体商业规则，同时融入 IT 信息系统的优势；
2. 通过电子化流程提升整体交互、管理效率，即将业务流程分步骤、分优先级信息化，提升管理效率；
3. 商家数据信息分层管理，即商家数据分层管理，核心财务数据、关键业务数据、非关键业务数据分权限分角色管理，提升管理效率，进一步降低业务风险；
4. 丰富商家服务功能，即为商家提供更多的基础服务功能；
5. 发挥商家团队的整合角色优势，即发挥商家团队的独特优势，整合系统资源，为商家打造统一的经营管理平台。

确定了发展思路，后续的业务流程梳理、产品需求整理、开发工作也紧锣密鼓地展开了。通过与业务方不断讨论，最终确定以“商家入驻系统”作为第一个突破口。

2012 年 5 月底，依照整体系统规划确定的两个核心、3 种主要流程的长期发展目标，与业务方确定入驻系统整体需求，由 POP 平台 VP 作为业务负责人，POP 开发团队总监作为技术顾问，迅速开展后续工作。并在产品需求确定后，仅用不

到 4 周的时间完成开发测试工作并正式上线试运行。在两周的试运行期间，根据业务及商家反馈修改并完善了二十多个功能点，并于 2012 年 7 月中旬正式在京东前台开放入口。

商家入驻上线运行一年，入驻商家 10000 多家，有效地支持了业务的发展，但还是留下了一个心结，那就是——在线电子合同。没有在线电子合同，整体入驻效率就无法大幅提升。通过多次与业务、法务、财务、税务的沟通，终于制定了一整套电子合同方案。

2013 年年底，商家入驻二期开发团队整装待发，长途迁徙至“莫斯科”会议室进入封闭开发，又一次时间紧任务重的挑战来临，但是“电子合同”这个心结终于可以在每个人心中消失了，大家不禁要喊：终于实现了在线电子合同，并完成了与公司合同管理系统实时对接的任务。这个梦做了一年，但是实现它我们仅用了一个月，从 5×8 小时工作制变为 6×9 小时工作制，后续变为 6×12 小时工作制，最后 1 个星期甚至通宵联调测试，我们就是那一伙爱拼命的人。付出总是有回报，系统按时上线后，所有的运营数据都表明我们成功地走出了一小步：

1. 3 个月入驻商家总量就超过一期入驻商家总量之和；
2. 1 个月内完成 POP 几万家商家续签、缴费工作；
3. 单商家入驻平均时长由原来 15~20 天缩短为两天；
4. 商家资质数据实现电子化审核、存储、查询检索。

业务方曾这样给予评价，电子合同的实现，仅纸质合同邮寄费用每年就能为商家和公司节约几百万的成本，这也许就是我们团队的价值。入驻系统还有很多需要完善的方面，我们也一直不停地在改进。

团队人数由最初的 6 个人发展到目前的 14 个人（包括产品、研发、测试），回想我们共同制定的发展思路，还有很多东西未实现，后续整体团队会在商业规则、资质深度管理、平台商品质量监控、商家信息整合、业务流程优化、商家生态等方面重点推进，与京东一起发展进步。

## POP 是业务的开放，也是 API 的开放

2010 年 POP 平台刚开发的时候，就考虑到一些大商家有通过接口实现与公司内部 ERP 系统对接的需求，于是 POP 开放了自己的 API，当时 API 的调用形式是 Web Service 的。



Web service

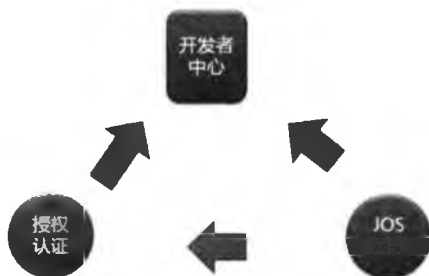
最早的接口开放

2011 年是 JOS 比较重要的一年。这一年 JOS 的用户开始逐渐增加。随着用户量以及没有开发能力的商家的增多，商家的需求也在不断地膨胀。基于这个现象我们开始产品化我们平台，我们希望能引入更多的 ISV 来开发满足商家需求的软件，建立开发者中心让开发者统一申请 App，建立 App 和开发者的概念。以前的开放平台没有授权的概念，只有商家 ID 作为唯一标识，为了让整个开放平台更加的安全，我们引入了 OAUTH2.0 作为授权标准。



OAUTH 2.0 授权流程

随之，我们开发了开发者中心和授权服务器，并改造了网关。3 个项目一气呵成，开放平台的产品迅速上了线，形成了一个开放产品的铁三角。记得当时我们的 VP 李大学给这个开放平台起了第一个名字 JDOS。后来感觉这个名字不够气派不容易记住，老李就改成了 JOS（宙斯）。



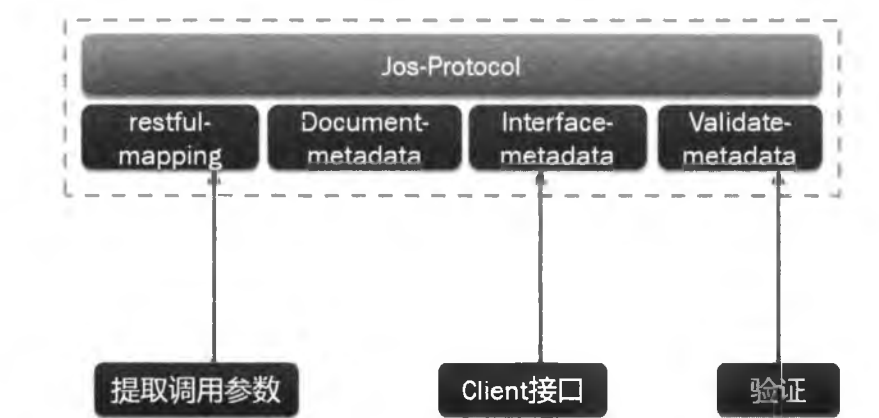
JOS产品铁三角

2012 年，随着开放趋势越来越明显，开发者对 API 的要求也越来越多。对于开放平台来说我们应该去增加更多的 API。老的开放方式是全部人工，我们 JOS 的开发人员需要人工将每个 API 开放出去，并且需要写文档 SDK 等。



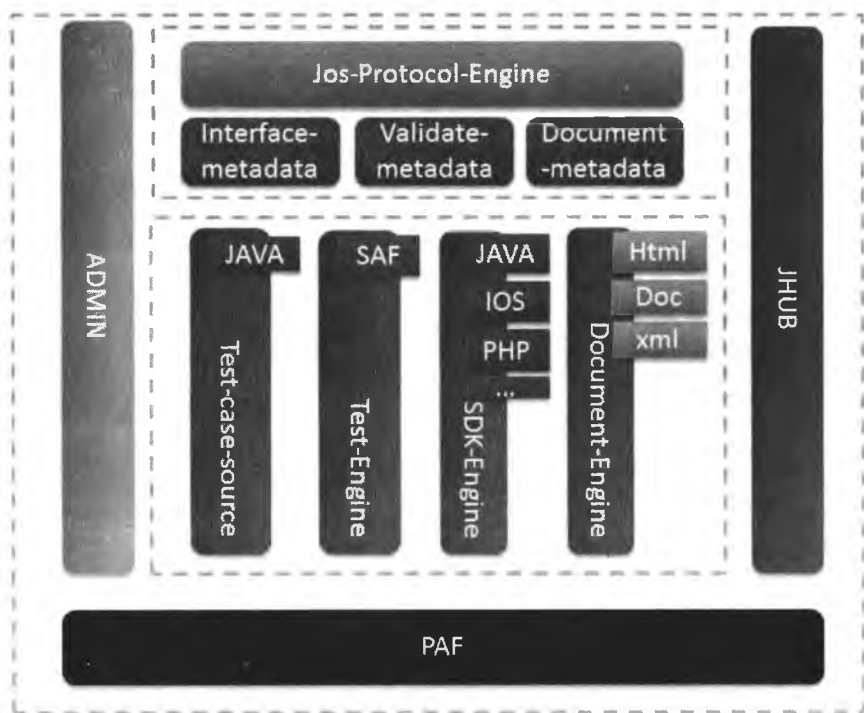
JOS 人工接入方式

我们还有很多痛点，比如，并行开放的 API 受制于 JOS 的人员限制、一个团队是无法了解整个公司的业务等。于是 JOS 的平台化就提上了日程，目标就是把 JOS 做成一个让内部的伙伴可以自己开放 API 的平台。我们建立标准、制定统一的接口描述规范 JOS-IDL，不管你是什么协议的，我们都可以进行转换和调用。



JOS 接口描述规范

我们还建立了服务接入平台 JHUB 来接入服务并自动化生成文档、测试工具等。

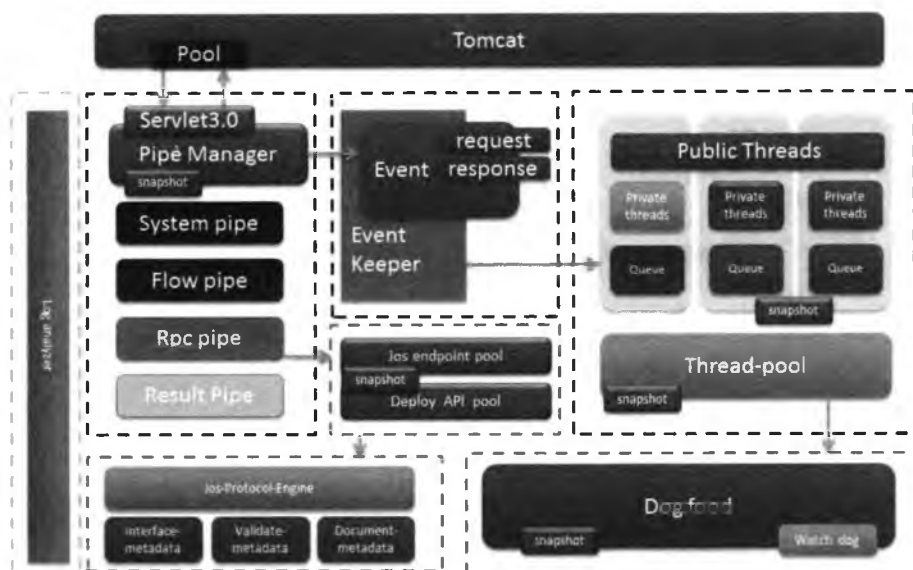


JHUB 架构

这样一来大大降低了开放 API 的时间。可以自动化地产生 API 文档、测试工具等。还可以并发地接入 API，让 JOS 的发展插上翅膀。随着数据量不断上涨，我们也构建了自己的离线计算集群和实时计算集群用于监控我们的系统。

2013 年，随着 API 开发者不断增多，新的产品销售渠道产生了问题。这时候，我们考虑是否可以通过建立一个服务市场来满足开发者和商家更大的对接需求。于是 JOS 开始搭建服务市场让生态变得更加繁荣。

随着调用量不断增大，我们意识到了调用的平稳容灾的重要性，于是我们重构了网关，建立了一个可用于 API 隔离的架构设计。首先是异步化，让请求的处理权完全从容器的控制交给我们控制，内部调用的管道化可以平滑动态地降级处理业务。API 作为资源调用后，我们改进了调度算法，用一种权重的形式代替了 FIFO，将 API 调用进行隔离。



请求异步化处理架构

2014 年是快速发展的一年。我们不断地通过实时监控获取调用情况，也通过实时分析获取性能情况。消息主动推送系统也正式上线了，进一步升级了我们的文档系统以更加友好地关怀我们的用户，不断完善数据并指导系统升级，在授权方面也更加细化和开放了 JOS 的授权按钮。

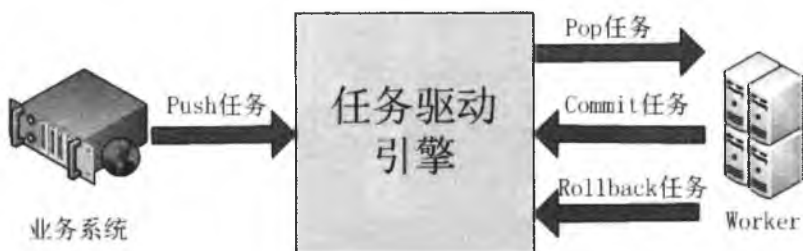
## 任务驱动引擎

2012 年年初，POP 系统所使用的 Oracle 数据库曾发生过一次宕机事故，原因是大量的 Worker 程序频繁扫描数据库的各类任务表，由于 AQL 过滤条件字段不适合建索引，导致 SQL 的执行基本上是在做全表扫描。部门决定让刚入职的李飏开发一套专门用于存储和管理任务的系统，任务驱动引擎就此诞生了。

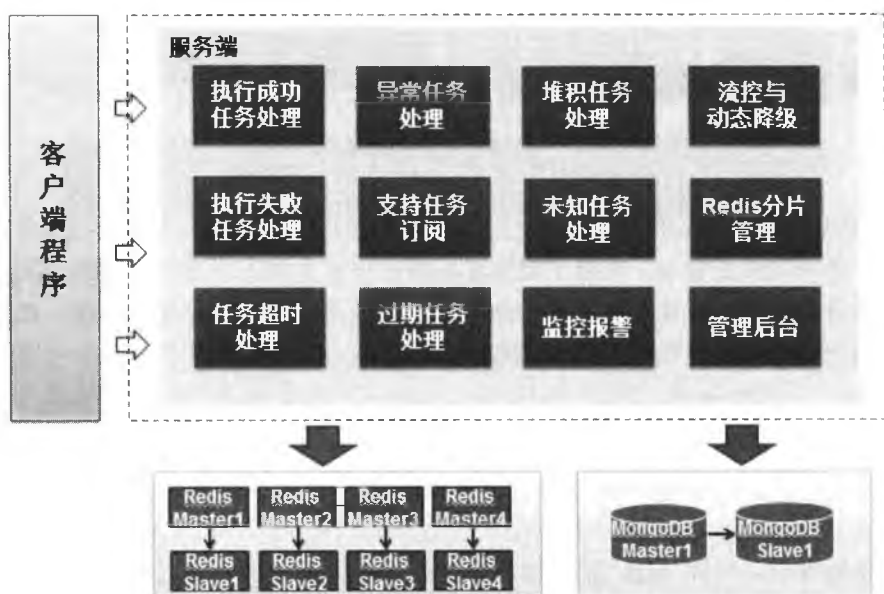
任务驱动引擎是基于 Redis 和 MongoDB 的任务存储与管理系统，提供高性能、稳定的任务存储与管理服务。从使用者角度来看，任务驱动引擎提供以下 4 种主要功能：存储任务(Push)、消费任务(POP)、消费成功后提交任务(Commit)、消费失败后回滚任务(Rollback)。

任务驱动引擎接收并处理客户端提交的以上 4 种请求，如下图所示。使用了任务驱动引擎后，任务是存储在 Redis 的待执行队列中。当任务被消费后，任务从 Redis 的待执行队列转移到正在执行队列中；当任务消费成功并提交后，任务会从 Redis 的正在执行队列中转移到 MongoDB 的已完成任务表中；当任务失败

并回滚后，任务会重新转移到 Redis 的待执行队列中或者转移到 MongoDB 的失败任务表中。



任务驱动引擎与业务系统交互示意图



任务驱动引擎功能结构图

任务驱动引擎从 2012 年 3 月上线以来，为 POP 系统的稳定性做出了非常大的贡献，所有的 POP 系统均接入了任务驱动引擎，目前，任务驱动引擎日处理任务数近一亿，为 POP 各系统提供了高性能、稳定的任务存储与管理服务。

## POP 新业务，托起 POP 的另一片天

POP 平台由于其开放平台的天然基因，对于新业务的包容性强，支持相对快捷，公司的许多新业务模式拓展都是基于 POP 平台展开的。另一方面，当时的 POP 业务负责人马总是大家公认的“新业务发明家”，业务创新如滔滔江水绵绵不绝，各种新业务，如团购、虚拟产品都是在马总的推动下实现的，POP 业务从此百花齐放。

### 团购之战

2010 年，团购行业发展如火如荼，上半年还是“百团大战”，到了下半年就演变成了“千团混战”。作为自营 B2C 行业的老大，京东自然不能作为旁观者，2010 年 11 月我们在苏州街研发中心的一个小角落里，接到了公司的紧急任务：圣诞节之前团购网站要正式上线，当时只剩下 20 天的时间。

当听到公司要迅速搭建团购平台开展团购业务后，POP 研发部成员马上讨论出了一个可行的解决方案。时间紧、任务急，大家之前都没有过团购业务系统研发经验，如何才能按照公司的要求在圣诞节前上线运营呢？想了半天最后只有一条看似可行的路：先用开源的软件改出来一个能够融入京东信息体系的系统。在紧张选品之后我们马上锁定了一款团购开源软件，并迅速进行采购等流程，将最原始的一份代码拿到手。随之而来的第二个问题就是程序是用 PHP 写的，但当时京东绝大部分研发人员都是 .Net 和 Java 选手，这一下又让大家挠了头。随后 POP 中的两位 PHP 业余选手薛韬和韩笑跃决定临阵抱佛脚来博一把看看能否搞定这个系统。决定了人员和时间节点后，杨凯带领着这两位业余选手开始了对团购系统的改造开发工作。功夫不负有心人，他们只用了不到 20 个日夜就将京东的用户系统、订单系统、支付系统、商品系统、结算系统逐个打通，将这个买来的程序完全融入到了京东信息体系中。第一个团购项目于 2010 年 12 月 14 日上线，第一个订单成功支付并发放优惠券时忙得天昏地暗的 3 个人总算是可以歇口气好好休息一下了。

### 游戏点卡

随着 POP 业务步入正轨，业务量增长迅速，商品种类多元化，为了满足商



城的发展需求，为用户提供更好的一站式服务体验，为商城带来更多的营业额，我不需要拓展新的商品种类，我们应该引入虚拟产品到商城中。虚拟业务的特点在于下单到购买成功实现线上完成，无仓储配送环节，交易快捷简便，一个订单只赚五毛钱那也是实实在在的利润。

以下为游戏点卡项目负责人杨凯的回忆。

“游戏卡密/直充”是 POP 的第一个虚拟业务，我很荣幸当时有机会主导研发“游戏卡密/充值”项目来支撑公司第一个虚拟业务的开展。

当初在系统设计过程中有 3 个问题令我比较纠结，从 3 年后的今天来看当时的选择还是不错的，在此分享一下当时的几个困难。

### 1. 选择虚拟业务的商品不入京东商品库。

2011 年 2 月以前，京东的每一件商品都出自京东商品库，但虚拟业务的商品和实体商品比起来有很大的差异。虚拟商品没有京东商品的一些必备的商品数据，商品内容灵活多样，京东商品库的结果很可能无法满足，今后需要不断地改造。虚拟商品的数据量无法按真实 SKU 的经验来评估，如果虚拟商品进入京东商品库，也存在系统隐患。在这种情况下，我们决定自己建设虚拟的商品库，自己做商品的详情页和列表页，从后期来看，这样我们可以方便地接入各个虚拟服务商的虚拟商品，也可以“随心所欲”地设计当前业务的网站体验，颇有轻装上阵、放手一搏的感觉。

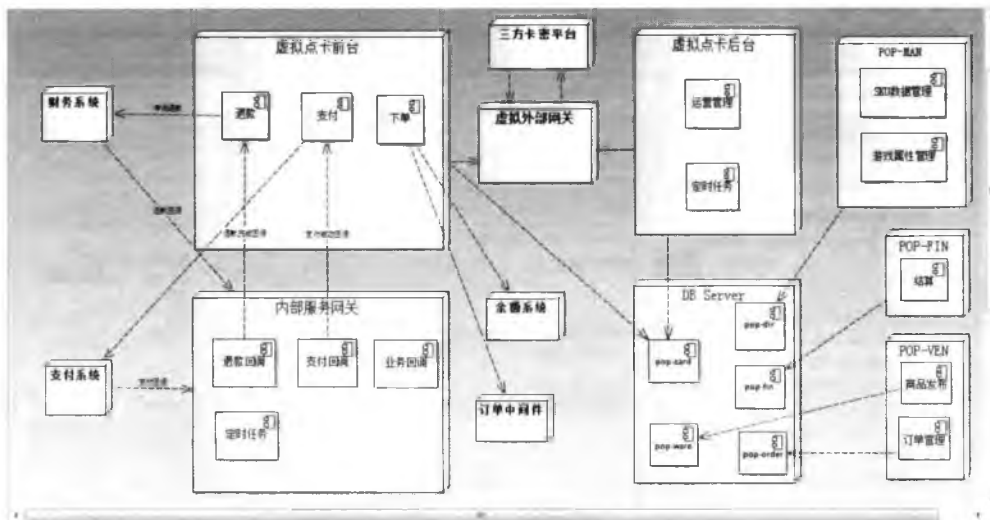
### 2. 决定虚拟业务订单和京东 ERP 订单两条腿同时走路。

公司的财务结算直接依赖京东 ERP 订单，但是京东当时的订单状态无法满足虚拟订单状态，比如，游戏卡密充值中存在“充值中、充值成功、充值失败”这样的状态，今后其他虚拟业务，比如，彩票，机票有各种订单状态，放在京东订单上太困难了。经过分析后每种虚拟业务都有自己独特的订单状态，我决定来建设虚拟业务的订单系统满足业务的个性化需求，京东订单同时存在，主要用于在线支付后的对账，也利用 POP 商家结算系统给虚拟服务商结算，这样用户和商家的需求同时都满足了，也不需要订单、财务、结算系统有任何改造，当时的这个方案使整个项目组松了一口气。

### 3. 建立网关系统，统一面向未来所有虚拟业务的支付和财务服务。

因为不存在配送环节，多数的虚拟业务在线支付需要依赖京东的在线支付体系，售后也需要依赖公司的售后退款系统，在这些业务中都存在这一种叫做“反向实时通知”的场景。比如，支付成功后在线支付系统需要告诉游戏充值系统，

退款财务审核成功后也需要通知游戏充值系统。今后彩票、机票、手机充值等业务都存在这样的业务场景，我们无法要求财务系统，售后系统都能配合我们做改造，所以我们建立了反向通知网关，只要是虚拟业务所有的反向实时通知都通知到网关，网关决定通知到具体的哪个虚拟业务系统，这样把可变的部分都控制在自己手里，真正达到了高内聚，低耦合的设计要求。



游戏点卡系统架构设计

今天，虚拟游戏充值业务做得比当初规模大了很多，但是从这第一个虚拟业务系统研发的过程中我们所积累的设计经验一直为后期所有虚拟业务产品线服务着，也许这就是通过技术带来的价值。

## 京东手机充值

2011年3月，确定POP研发部负责手机充值系统研发，当时POP研发部资源非常紧缺，而且POP平台刚上线不久，业务发展和业务需求都处于爆发期，占用了绝大部分开发资源。彪哥综合考虑后，决定让丁琼负责手机充值系统的研发，调来实物组刚入职的陈保安同学负责商家充值接口对接，游戏点卡组张坦同学负责充值前台功能开发，刚从成都研究院来北京培训的涂浩同学进行充值后台功能开发。4个人的开发团队迅速组建完毕，大家积极配合产品经理梳理业务流程，完成系统原型设计，2011年4月底确定了充值系统最终版PRD。需求评审完成后，大家立刻开始了系统设计，2011年5月中旬完成了设计评审，2011年6月初完成了整个系统的开发工作，2011年6月底完成了系统测试，并完成了线上

应用和数据库环境搭建，2011年6月23日完成预发环境测试，经过一周内部试运行，整个充值流程全部走通，2011年7月1日新充值系统正式对外。新系统正式上线后京东充值速度和稳定性都有了大幅提升，以下为新系统订单中心效果图：



手机充值订单中心

## 京东彩票

继京东充值项目开工后，2011年5月彩票的需求也随之而来，彪哥又将当时负责 POP 计费的管亮祺抽调过来和大家一起负责彩票系统的研发，经过兄弟们两个月的奋战，2011年7月彩票系统成功上线，从此翻开了京东彩票的篇章。



彩票频道页面

2012 年初，为了扩大用户群体，我们决定增加“快开型”玩法，10 分钟一期，一天 84 期开奖，确实很吸引一批“急性子”的用户。经过了一个半月的需求调研、开发、测试，“新时时彩”玩法于 2012 年 3 月成功上线，但是销量并没有达到预期效果，每天“新时时彩”的订单销量惨淡，然而值得高兴的是“数字彩”的销量确在节节攀升，看来虽然新增彩种上线没有体现出订单量的增长，但却因丰富了京东彩票的彩种，吸引了新的用户，在一定程度上说明我们更加成熟了。



## 京东特卖之路

随着 POP 开放平台规模的壮大，需要一些新的业务形式来变现剩余流量，提高销售量，所以从 2012 年开始，POP 陆续开展了海外购、卖车、卖房、拍卖等业务，且都已取得了非常好的销售业绩，同时也引起了业内关注。在这些新业务发展的背后，离不开技术力量的支撑。提到支持 POP 新业务，不得不提到珊珊，每项新业务的技术实现过程中几乎都少不了她的身影。珊珊是典型山东豪爽女，对于做项目这件事，她的口头禅是：虱子多了不痒，债多了不愁。以下是对她采访之后的整理编撰。

### 闪团与闪购的前世今生

2012 年年底，团购从 POP 独立出来成立一级部门，需要更大的战略来调整团购业务的组成，经过一系列高层磋商，在 2013 年年初就诞生了“闪团”这项业务，定位正品特价，限时抢购，仓储部门单独在成都成立了一个仓库作为闪团仓。

在项目初期，界定项目的范围花费了好一番时间，FBP 商品、SOP 商品、LBP 商品、SOPL 商品、京东自营商品都有入仓进行闪团的需求，那时候还没有“开放平台仓”这个明确概念，只有 FBP 和自营商品能够入仓，所以敲定了一下如果是其他模式的商家，需要单独开一个 FBP 的店铺。而自营商品因为与 POP 的结构不同，一期也先不作为重点考虑对象。集中精力先把 FBP 的入仓模式搞定。

FBP 的库房属性又让人很纠结。因为原来商家的商品都是在 6 地备货的，每地仓库都有不同的覆盖范围。而闪团是要求在成都一地备货，能够支持全国发货。那时候刚刚开始搞平行库存，还没有成熟的出库优先级的模型，同一个 SKU 入了闪团仓就不允许同时入普通仓，否则会造成发货混乱。并在源头就卡死这种同时入两类仓的可能性，要做闪团的商品只允许入闪团仓，需要店铺要新发布 SKU 来参加闪团活动，然后在发布商品上有个标记来区分，有这种标记的商品就直接绑定闪团仓的库房属性，无法再下到其他 6 地普通仓的入库单据。

其他纠结的点还有很多，比如 30 分钟取消订单、逆向流程、确定闪团是以促销的形式发布。因为会议室难找，二层办公楼一间空闲的 VP 办公室就被暂设为“闪团作坊铺”，当项目经理和产品经理需要讨论作战计划时就到这个小屋里

[illegible]

## 閃團系統流程

这个流程图的背后是将近 20 个系统的梳理和修改，其实现在再来看这套方案确实不算什么了，并没有太复杂，但是当时的情况下确实是趟了很多的冤枉路、讨论了很多细节的东西，才确定出了这套方案。

2013 年 5 月 22 号，上线的前一天晚上，项目组主要成员以及 WMS 的几个研发人员一直奋战到凌晨 3 点才把问题彻底解决掉。第二天闪团终于如期上线，也卖出了不错的业绩。遗憾的是后来随着团购部门战略调整，到 2013 年 10 月闪团业务运营了 4 个多月之后被暂停了。

但是京东并未放弃这项业务，2013 年年底，由 POP 部门再次提出并启动，而此时不管是在仓储开放平台仓的准备上（是的，这时候已经有开放平台仓的概念了，可以支持三方卖家入京东仓。北京，上海，广州三个仓作为开放平台仓），还是在供应商的选择上，时机都要更加成熟。于是轰轰烈烈的“闪购”项目又开启了。为了拉动过年前后的销售低迷期，业务部门希望在一月初能够将闪购正式推出。而那时距离一月初只剩一个月时间，时间非常紧迫。

系统设计上第一个关键问题是整体方案用 FBP 实现还是用 SOP 加开放平台仓来实现：FBP 是现成的入仓模式，但是不够灵活，商家的自主性低，而 SOP 加开放平台仓是一种新模式，预估开发量会比较大，存在未知风险，但是却更加灵活，也是未来的方向。经过研发部好几个部门的负责人加产品经理多番讨论之后，最后敲定使用 SOP 加开放平台仓来实现。

大方案确定了，项目迅速开动。项目的推进过程分为 3 大部分：第一部分是前端页面包括频道页、活动页、详情页等，第二部分后台系统流程和各环节改造梳理，第三部分是对接仓储开放平台仓服务，分别有产品经理重点跟进。

因为有着之前做“闪团”的经验和系统基础，珊珊这次感觉要省力得多，套路是相似的，只是在后台系统流程设计上比较头疼。闪购业务用的都是京仓和京配，那么就有两种方案：一是开通闪购业务的商家使用一个标记来标识，根据这一个标记来识别后续订单售后等不同流程；二是把闪购、京仓、京配分别用 3 个不同的标记标识，后续流程根据不同的标识各判断各的。经过好多次的讨论，由于闪团是基于 SOP 的业务，未来 SOP 加服务的模式服务之间一定要彼此解耦，所以用 3 个标记分别标识是更合理的。为了方便运营操作，系统做了一个联动，在开通闪团时自动开启京仓和京配服务，等到闪购与京仓和京配独立的时候，把联动去掉就可以了。（事实上没过多久，这个需求就提出来了）。

然后此项目最重点的是与仓储对接这部分，入库和退库因为有 FBP 的参照，相对来说还好一些。库存和订单的对接着实多费了一番周折，同时还需要架构师



的介入。由于时间紧，所以很多方案都是边开发过边敲定的。逆向流程根本没有时间仔细梳理，尤其是发起退款锁定订单这条，按照 SOP 的流程，订单被锁定是无法出库的，但实际仓库操作过程中复核之后就无法拦截了，所以复合之后即使系统不能出库，包裹也无法归还。这是一个比较大的流程上的漏洞，但是为保整体项目进度，我们只能先把审核权限放在了 Man 端，由人为线下沟通之后再审核，报了多次项目风险之后，先把项目上线了。



闪购业务“京东红”上线

2014 年 1 月 8 号，闪购业务正式上线。它还有个好听的名字叫“红”。而产品经理们并没有放松下来，因为还有逆向退款流程必须要解决，于是项目经理召集了二十几号人在一个会议室里关于这个问题进行了深入讨论，最终确定退款申请由财务系统发起推送到仓储开放平台（JOSL），由 JOSL 询问 WMS 是否可以拦截，如果可以拦截，退款单自动审核通过，如果不能拦截，JOSL 会再询问配送，由配送反馈是否可以拦截。按此方案各系统紧急开发上线之后，大家才算真正松了一口气。

闪购也没有辜负大家所望，越做越壮大。2014 年 7 月，闪购已独立为一级部门，成为京东的第八大采销，队伍也在不断扩张，甚至也已经有了单独的产品团队，后面的发展肯定会越来越顺。

## POP 生态建设新时期

进入 2013 年下半年，伴随着京东在电商领域的大踏步前进，京东 POP 业务也进入了新的时期，POP 系统也迎来了许多新的模式，包括商家奖惩系统、店铺动态评分系统、闪购业务、分销业务、LOC，以及京东和腾讯合作后的第一个合作项目——微店项目。这些新模式，既对 POP 业务进行了扩展，也将 POP 的生态建设得更规则和平衡。

与此同时，由于前期疲于应对业务需求，加上电商业务偏逻辑的特点，整个 POP 研发团队对于卖家体验重视得不够，与卖家的接触也不多，整个过程缺乏连贯性。为了更好地服务卖家，提升卖家在京东的开店体验，从 2013 年下半年开始，POP 研发团队多了另一项重要职能——提升卖家体验。

以下是徐建文的回忆：

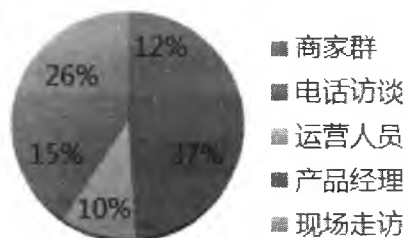
记得那是 2013 年 8 月 13 日，我刚刚入职成为京东的一分子。白天办理完各种入职事项和入职学习，以为入职第一周无论如何也会比较轻松。

可是上班第一天就给我一个 Surprise。晚上七点，领导叫我参加一个什么会议，我也没听清楚，就急匆匆地从二楼跑到了 16 楼。会议室里密密麻麻挤满了人，在挨个汇报自己当天的工作完成情况和第二天的工作计划。后来我才知道，这场腥风血雨的战斗有一个霸气的名气“雪耻行动”，而我们分属于卖家战区。

在接下来的一个多月时间里，每天晚上七八点，几十号人聚集在一个会议室总结当天的战况。而在会议室外，有几百号人在为这一次战役奋斗着。这些人被分成 16 条产品线，各自找出自己的产品问题，并作出改进计划，改进的地方通过各种方式通报给商家，逐步让商家试用并提出改进建议，各产品线再进行持续优化。

雪耻期间，我们共访谈商家 265 家，其中大型商家占 70%，中小商家占 25%，新商家约占 5%，收集原始需求 396 项，排重分期处理后，确定雪耻需求 304 项。

需求总数	396个
商家群	49个
电话访谈	146个
运营人员	40个
产品经理	58个
现场走访	104个



需求来源统计

我们投入了产品 92 人，研发 276 人，测试 42 人，设计 3 人，专家 1 人，围绕下面 5 个核心方向展开工作。

1. 可用性：对抱怨较多的功能点的优化；
2. 效率：简化操作流程，优化交互界面；
3. 精细化运营：更多的功能支持；
4. 服务：强化各类帮助，初步将商家反馈方式体系化；
5. 卖家生态：健全更为公平的商家竞争环境。

到 2013 年第四季度，商家反馈良好，对京东充满信心。这也给我们吃下了一颗定心丸，只要肯付出，就一定要有收获。

## POP 未来使命与荣耀并存

从 2010 年 3 月 POP 项目启动、平台规划，一直到 2014 年 8 月，京东 POP 平台走过了 4 年多的岁月。这是一首高歌猛进的创业进行曲，虽然中间多有曲折，但 POP 平台取得了长足的发展，为京东商城提供了多样化的品类，也是迄今为止京东最大的开放探索。京东自主建设的 POP 系统也经历了多次大的架构升级，成功抗住了一年又一年爆发式的增长。POP 系统的建设为京东 IT 事业提供了非常多的最佳实践案例和经验教训，也为京东研发各个系统输送了有丰富电商实践经验的研发、产品人才。我们必须向 POP 平台自规划以来所有的建设者致以最崇高的谢意，不论你们现在是否还在京东，POP 的成功都有你们的贡献！

随着京东成功圆梦上市，POP 平台必将迎来更广阔的舞台，同时也必将面临更进一步的挑战。机会与挑战共存，这是每一个上市公司都要面临的境况。而与腾讯的合作，将是有机会改写中国电商历史的契机，POP 研发部门在这当中将会承担更大的作用和责任。POP 平台将继续秉着创业团队的精神，将 POP 平台建设得更好，为京东电商开放事业增添新的光辉！（作者/王彪、欧阳波、谭悦娴）

# 8

## 速度与激情——极致

## 体验的供应链

### 仓储系统发展历程

#### 京东 WMS 发展的第一次探索——上海老 A 库版本

虽然京东仓储以前存在过 WMS1.0 系统，但当时那套系统流程非常简单，也没有智能化设备，采用集中部署模式，依赖在线生产模式，能够承载的单量也极为有限。

2009 年，公司着手进行新一代智能化仓储系统的研发。当时，4 名研发人员在经历半年多的封闭开发后，于 2009 年 11 月让这套当时看来是“新一代”系统的上海老 A 库系统正式上线。经历了一段痛苦期后，当时的研发人员最终为京东研制出了第一套接入智能化设备的全新 WMS 系统。该系统为京东首次接入了智能拣货手持终端、智能输送线等设备。相对于当时的 WMS1.0 系统而言，它不仅是一套“高大上”的系统，而且也极大地提高了准确性和生产效率。虽然系统能够承载的单量比较少，系统升级的潜力也不大，更不受现在的维护人员的青睐，但对于京东 WMS 发展来说，它是有着重要历史地位的。

## 京东 WMS 的生态链构筑者——WMS2.0 系统

在研发完成上海老 A 库系统并应用实施后，公司尝到了新系统所带来的甜头，决心研发一套通用型标准化的京东 WMS 系统，该系统后来被正式命名为 WMS2.0。京东 WMS2.0 相对于 WMS1.0 系统实现了业务和技术上的质的飞跃。系统扩展了大量的业务，这些业务不仅具有很高的完备性，而且成为了京东仓储后续 WMS 版本的业务范本。

京东 WMS2.0 系统有完善的入库、在库、库存、出库、内部配货、逆向物流等模块，每个模块下又细化划分出了如出库的定位、任务分配、拣货、复核、打包等子模块。这些业务上的功能节点，形成了京东自己的一套 WMS 的业务风格。京东后续开发的几版 WMS 系统的主要的业务流向，依然参照 WMS2.0 的业务节点，在此基础上或是重新设计架构，或是调整功能。

京东 WMS2.0 系统的业务流程，其实并不是从开始就设计完备的。从 2010 年上线到现在，几乎所有的模块都进行过重新开发，有很多业务模块都是后来增加的。这份成果离不开京东大量的单量验证总结、仓库一线工作人员的使用反馈、系统的开发人员无数次夜半远程库房解决问题，以及需求收集后的重新开发上线。发票打印经历过在线打印到离线打印的切换；面单、集合单经历过集中打印的改造；内部配货经历过多版设计及重新开发；库内的移库补货经历过由人工判断到智能补货的进化；任务分配、复核都重新设计和开发过；单据生产数据及状态回传经历过多次架构重新设计；入库能够做到了流水作业……局部模块的重新规划设计，生产流程的多次设计开发，这些年对 WMS2.0 的改造，让这套系统为新一代的 WMS 系统提供了良好的业务参照。而且，京东后续开发的 WMS 大家电版本，以及协同仓版本的 WMS，都是直接在 WMS2.0 之上进行需求定制的二次开发。WMS2.0 并非最优秀的仓储系统，却撑起了目前国内最大的仓储生产单量，而且暂时还未达到生产瓶颈。

## 京东 WMS 的第二次探索——WMS3.0 系统

2012 年，公司决定开始要用新的架构设计理念去研发新一代的仓储系统。架构设计上采用 SOA 的服务化设计思想，并引入任务引擎来处理消息，让 WMS 系统各模块、各子模块之间的数据，可通过服务调用的方式来进行交互。这样的架构设计，能够解决 WMS2.0 单仓部署成本高、数据存储不集中、不易提取数据等缺点。因为当 WMS 各模块服务化之后，公用的模块可以进行集中部署，多个单量小的仓可以集中部署，单量大的仓可以单仓部署。此外，模块服务化之后降

低了各模块之间的依赖，方便了模块的定制。

但是，对于当时的京东仓储团队来说，做这样一件事，是有很大风险的。原 WMS2.0 系统的需求改进仍然在不停地进行着，不能让整个团队都投入新项目的研发中去。对于新一代架构设计，在很多业务的实现上增加了很大的难度，项目研发中将会遇到的问题具有很大的未知性。

在这种背景下，被命名为 WMS3.0 的新一代仓储系统开始了研发工作。

果不其然，项目在研发过程中遇到了很多设计上的问题，从业务、产品以及研发等方面对需求设计进行了无数次修改，整个 WMS3.0 研发团队在进行了半年多的迭代开发后，最终让这套系统在北京的 98 号测试库上线了。后来公司决定在这套系统上进行接入开放平台订单的业务探索，验证京东仓储由开放平台接入第三方订单，进行第三方订单的单独生产的多货主业务模式的可行性。

WMS3.0 系统并没有大规模推广，原因主要是从设计开始，WMS2.0 也在不停地进行业务迭代更新，导致 WMS3.0 系统的业务功能始终是落后于 WMS2.0 系统。随着亚洲一号系统的上线，仓储团队产生了线上同时运行 WMS2.0、WMS3.0 和亚洲一号 3 套仓储系统的局面，故而有了后面为整合多套系统业务而诞生的 WMS5.0 系统。

WMS3.0 系统虽未全国推广应用，但还是有着很大的意义。京东亚洲一号的仓储系统之所以能够较快地研发，是因为亚洲一号的 WMS 是在 WMS3.0 基础上进行的改进和需求调整。

## 京东 WMS 的新飞跃——亚洲一号系统

尚未加入京东时，Kun 就已经听说了京东要建造自己的高度自动化生产的库房——亚洲一号。至于项目占地面积，库存量等等数据，网络上随处可以查到，我就不在此赘述。

这套系统在组成部分上其实是分为 WMS 和 WCS 两部分，京东在 WMS 领域已经驾轻就熟了，但是 WCS 系统对于京东来说还是空白。

何为用到 WCS 系统？在京东亚洲一号库房中，除了需要传统的 WMS 管理系统外，还引入了大量的提升机、堆垛机、智能输送线等自动化设备，这些设备本身有自己的驱动程序，但是如何将 WMS 的生产消息转化来驱动智能设备按照京东 WMS 的命令进行工作呢？WCS 就是完成这项工作的一套系统，充当着京东 WMS 系统和智能设备驱动命令间的翻译官，需要将 N 多种 WMS 业务命令转化

为设备或设备间需要做的动作。相关联的，设备的动作执行的异常处理方式，设备是否正常工作的管理机制也都包含在内。

亚洲一号系统，并不仅仅是在传统 WMS 中加入智能化设备这么简单，既然增加了 WCS 系统，那么 WMS 的各个模块就要与 WCS 进行消息通信，这种通信机制的建立是对于现有 WMS 系统一种扩展。此外亚洲一号系统的 WMS 还进行了很多为适应自动化生产而进行的业务变动，系统的出库等模块设计与 WMS2.0 相比，业务上有较大的发展，使之能够支持“一件一包裹”的生产模式。

随着亚洲一号系统上线并顺利接入单量生产，京东仓储的发展已经到达了一个全新的高度。

### 京东 WMS 的全面开放——WMS5.0 系统

亚洲一号系统、WMS3.0 系统、WMS2.0 系统，京东的库房存在着 3 种不同的仓储生产系统，这给维护者来造成了很大的负担。

恰逢公司决定要对京东仓储进行全面开放的时机，公司决定对多套系统进行整合，整合出一套通用标准化 WMS 系统，系统的开发基础在上海亚洲一号 WMS 系统基础上进行。

系统的研发目标是，在将来取代目前线上的所有 WMS 系统，其业务功能点最全，并且最重要的是要实现促成项目启动的目标——京东仓储全面开放。

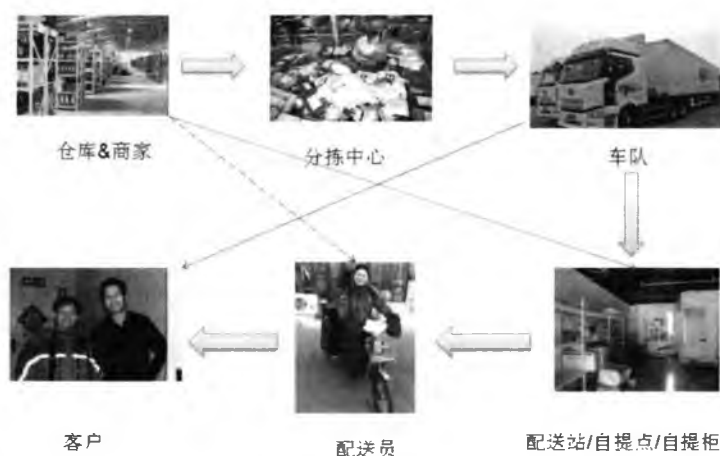
全面开放是在 WMS3.0 试验的仓储开放的业务基础上进行，能够支持京东仓储的库存、生产满足多货主的管理模式。这对于京东降低运营成本、增加收益来说有着决定性意义，也对促进整个社会的物流发展水平来说有着极大的推动作用。因为京东仓储的发展水平大家是有目共睹的，所以吸引了更多的第三方货主来体验和使用到京东的仓储生产服务。这对于京东来说也是一种提升影响力、展示仓储实力的表现。此外，仓储的开放能够使京东库房的生产产能得到平衡，为京东提供另一种盈利途径，毕竟仓储的开放是要收费的。



## 物流配送系统发展

电子商务的价值在于通过低成本高效率的运营，减少中间流通环节，为客户带来实惠的同时实现企业自身价值。电子商务的信息流、物流、资金流一直以来都是人们广泛重视和研究的要素。物流虽然只是商品交易的一个组成部分，但却是商品和服务价值的最终体现，要求以尽量低的成本将商品在正确的时间送达正确的地点，降低流通成本，提升供应链效率。“以客户为中心”在物流上最终体现，就是“最后一公里”的服务品质，这也是京东核心竞争力所在。京东正是依托多年打造的物流配送体系，让消费者充分享受了“足不出户，坐享其成”的便捷。

在京东的物流网络中，核心要素包括仓库、分拣中心、配送站、配送员。仓库负责根据客户订单安排生产，包括面单打印、拣货、发票打印、打包等，其生产成果体现为一个个订单包裹的生成。仓库生产完毕后，将订单包裹交接给分拣中心，分拣中心收到订单包裹后进行分拣、装箱、发货、发车，最终将包裹发往对应的配送站。配送站进行收货、验货交接后，将包裹分配到不同的配送员，再由配送员负责配送到客户手中。在整个配送网络中，物流、信息流与资金流的快速流转，实现了货物的及时送达、货款的及时回收、信息的准确传递。



京东物流配送主流程拓扑图

## 青龙系统 1.0

随着业务范围的扩大和业务量的增加，京东原有的物流管理系统已不能适应业务扩展的需求，构建更合理的业务流程和更高效的管理信息系统，是解决京东物流和信息流瓶颈的根本出路。

经京东技术委员会确认立项之后，项目于 2011 年 9 月 5 日正式开始启动。最初项目代号为阿拉丁，后来接受刘总提议，重新命名为青龙，这个名字寄托了公司对于这一核心系统的厚望。（青龙是中国传统文化四象之一，在四神兽中居首位，根据五行学说，它是代表东方的灵兽。京东之“东”为八方之首，而青龙为四神兽之首，预示着京东与青龙的紧密联系。）

从需求的提出、产品的设计到系统的研发测试，配送项目规划部、信息部的产品部和研发部 3 个团队密切配合，经过半年多的艰苦努力，完成了青龙系统基础架构的开发和测试。

### 2012 年 4 月 20 日

尚处于开发阶段中的青龙系统第一次上线发布，这次上线主要是进行基础功能试运行，包括功能测试分析、性能测试分析以及系统部署方案等其他细小却重要的方案测试。

针对这次的测试结果，项目组成员及时总结，提出改进意见和新的需求，力求系统功能的完善。根据第一阶段测试结果，开发人员紧接着进行了第二阶段的完善开发，经过数个日夜的奋战，青龙一期的功能全部开发完成。

### 2012 年 6 月 30 日

一期全部功能准备上线，开发人员对各个子系统进行细致的检查，灰度部署，最后确定线上验证方案。2012 年 7 月 9 日，青龙各个子系统在灰度环境下正式切换。2012 年 7 月 9-10 日，制定正式环境部署计划。同年 7 月 11-12 日，以华中区武汉 5 个站点作为切换试点。项目组成员对切换后站点反馈的问题进行汇总，讨论研究相应的解决方案。

### 2012 年 7 月 24 日

这一天注定是很多开发人员的不眠之夜。华中所有分拣中心和配送站点全部切换青龙系统，这一次也被定义为一场生死之战——这是青龙系统的首次正式亮相，也是对青龙项目所有参与人员无数个日夜奋战成果的一次严格检验。确保平稳，主流程操作正常，不允许出现 10 分钟以上的宕机，是这次系统切换的要求。

为了确保切换平稳顺利，项目组成员做了充分的切换准备，制定数据迁移方案、详细的 PDA 安装方案、日志调整方案、性能监控方案以及切换总体方案。

为了不影响正常生产，切换工作选择在深夜进行。前往现场指导的开发人员与集团总部的开发人员严格按照既定方案，密切配合，之后便是焦急的等待。凌晨时分终于到来，分拣中心和站点开始使用全新的青龙系统进行生产操作，结果令人振奋，青龙系统切换顺利，没有出现影响生产的重大问题。虽然期间分拣中心和站点都出现了一些问题，但是在开发人员的努力下都得以快速解决。青龙系统在华中低调而华丽的亮相，为全国各区域的系统切换奠定了基础。

### 2012 年 8 月 14 日

华南实现青龙系统切换。全国订单量较大的华南正式使用青龙系统，这是对青龙系统性能的又一次严格考验。开发人员严阵以待，充分的准备保证了系统切换的平稳。

### 2012 年 9 月 7 日

西南切换青龙系统，有了前两次的切换经验，系统切换平稳顺利。

### 2012 年 9 月 18 日

华东成功切换青龙系统，系统切换平稳顺利。

### 2012 年 10 月 23 日

东北成功切换青龙系统。

### 2012 年 11 月

青龙降临华北。至此，全国各区域的系统切换终于完成，青龙项 1.0 如期完成，实现了京东物流配送系统的大升级，为配送业务全流程、精细化的管理提供了有力保障。

青龙系统 1.0 实现了预分拣、货物交接、分拣理货、发货、验货、司机配送、站点收货、验货、配送员收货、送货，以及上门换新、取件、取件完成、订单入站、退货装箱等逆向物流的全流程管理。

为了支撑京东高速发展对配送业务的需求，青龙系统在设计架构上，对各个子系统功能进行充分讨论，即考虑各子系统分工明确同时兼顾各系统之间数据的共享，通过松耦合结构设计，为后续业务拓展提供支持。



青龙1.0核心子系统

## 基础资料

系统基础资料的合理设置对系统至关重要。青龙基础资料涵盖了分拣中心、配送站、配送员、司机、车辆、第三方承运商等基础对象设置，同时包括了商家合同、计费规则等复杂单据的设置。基础资料配置信息的合理设置，保证了系统对复杂业务场景的支持。

## 商家客户端及接货中心

青龙系统上线后支持京东配送承接外单业务，可以为任何需要发货的商家提供快递业务。

商家客户端提供了商家在线下单和批量提交订单等功能。商家可以通过客户端直接提交订单，也可以通过客户端跟踪订单配送进程，并处理后续代收货款的结算、运费结算等。

接货中心针对商家外单提供接货支持，接货完成后即完成商家货物到京东配送的交接。京东配送支持商家送货上门和京东上门取货两种操作模式，接货中心包括接货管理、运单打印、收货验货及退货等。

## 预分拣系统

预分拣系统根据收货地址等信息将运单预先分配到正确的站点，分拣现场依据预分拣结果将包裹发往指定站点，由站点负责配送。所以，预分拣结果的准确性对配送系统至关重要。青龙配送系统在预分拣中采用全址匹配、关键字匹配、GIS地理位置信息匹配等技术，保证了预分拣的准确性和高效性，为后续执行环节的高效运作提供保证。

## PDA 系统

PDA 系统是青龙系统中核心的生成系统，是分拣中心、配送站等现场操作的最强有力的利器。青龙 PDA 系统依据应用场景又分为分拣系统、配送系统、3PL 系统、合作站系统等，全方位支持京东配送各种业务场景的需要。

青龙为配送员提供 POS-PDA 一体机系统，在一台机器上同时实现了 PDA 和 POS 机收款功能，改变了配送员以往一手拿 POS 机一手拿 PDA 的作业模式，极大提高了配送员的作业效率，得到配送员的欢迎。

## 运单系统

运单系统既能记录运单的收货地址等基本信息，又能接收来自接货系统、PDA 系统等系统的操作记录，实现订单全程跟踪。同时，运单系统对外提供状态、支付方式等查询功能，供结算系统等外部系统调用。

## 质控平台

质控平台针对业务系统操作过程中发生的物流损等异常信息进行现场汇报收集，由质控人员进行定责。质控系统保证了对配送异常的及时跟踪，同时为降低损耗提供质量保证。

## 监控和报表

监控和报表为管理层和领导层提供决策支持，青龙系统采用集中部署方案，为全局监控的实现提供了可能。集团可以及时监控各个区域的作业情况，根据各环节顺畅度及时做出统筹安排。

## 财务结算

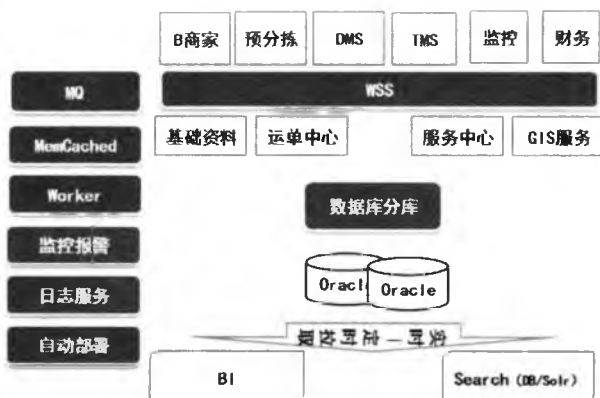
青龙财务结算提供了针对第三方承运商的运费结算和针对外部商家的运费结算以及代收货款的结算。京东每天有近 10 万单委托外部承运商进行配送，以前这些商家运费的结算全部依赖于手工 Excel 表的计算，工作量大、出错率高。青龙财务系统实现了运费的计算和结算全流程的系统化管理，保证了计算的准确性和流程的全方面监控。

## GIS 系统

青龙 GIS 系统在原有实现订单路径跟踪和站点信息展示的基础上，探索更深层次的应用，包括地址地理信息采集、配送区域划分、GIS 预分拣技术探索、单量 GIS 统计分析和配送区域优化划分设置等。探索通过对 GIS 系统的深度应用为

配送网络的优化提供支持。

青龙系统 1.0 是利用 Java 技术实现的 SOA 大型信息系统，在研发过程中，技术团队解决了大规模异步系统的构建技术难题，实现了灰度部署，服务柔性控制，统一监控和统一日志等。



青龙 1.0 技术架构图

### 性能与效率的提升

青龙系统 1.0 解决了信息系统不完善和系统架构不合理的问题，构建了更为合理的业务流程，同时也建造了更为高效的信息管理系统，还实现了海量信息处理的功能，完全能够满足海量数据的日常处理能力。青龙平台对原有系统进行了重构，使得分拣系统与配送系统达到了全方位的服务能力的提升。同时，通过提升系统运行效率，提高了站长、配送员等员工的工作效率，并支持京东销售规模的不断扩大，为用户带来了更为方便、快捷的购物体验。

### 业务种类多样性

青龙系统 1.0 全面地支持京东业务的多样性，不仅支持自营模式下的站外、站内及自提的配送业务，还增加了非自营模式下的合作站内、合作站外的业务类型的支持功能。青龙系统具备支撑未来京东快速开展二三线城市的自营站点与合作站点建设的能力，能够让更多的客户能够享受到“211 限时达”服务。

### 全面、精细化的信息管理

青龙物流平台功能全面，不仅实现了站点收货、站点验货、配送员收货、配送等正向操作功能，还实现了上门取件、上门换新等逆向功能，为用户提供了更

为方便快捷的配送服务。此外，还能实现对物流整个环节的监控和精细化的管理。实时监控系统、基本运营和分拣运营监控、配送运营监控及合作运营监控是青龙系统的 4 大部分，分别从宏观和微观层面对区域各个主要节点订单状态的分布、操作环节的差异进行分析，为公司决策和运营提供依据，很好地满足了配送质控、分拣中心、自营和第三方的业务需求。青龙 GIS 定位系统能让客户能够实时地了解订单的最新动态。

当然，青龙 1.0 的优点远远不只这些，青龙系统实现了信息化、集成化、模块化的设计原则。首先，青龙建立了完整的、面向客户的管理应用新模式，保证了京东各个部门以及其他相关企业之间的信息共享与畅通。其次，青龙系统将业务相互关联的部分集成在了一起，以自动化的业务处理取代了人工操作。

## 青龙系统 2.0

随着青龙系统 1.0 全国完成上线和运行，压抑近一年的配送业务方众多需求喷薄而出，因此，从 2012 年年底青龙产品和研发团队开始进行 2.0 的规划和实施。

青龙系统 2.0 的主线是根据业务方需求，快速完善功能。因此，规划了车辆管理、调度中心、逆向物流、站点集货、订单截留、预约配送、周转箱、内部质量管理、返单管理、物料管理、业务决策报表、报表邮件推送等项目。在产品、研发团队的共同努力下，规划的项目都得到了很好地实施落地。

与 1.0 相比，青龙系统 2.0 功能变得更加强大。



青龙 2.0 核心子系统

## 车辆管理与调度

在配送过程中，车辆作为最重要的运输工具，提高车辆管理的工作效率，实施规范化、科学化、系统化的管理直接提升整个配送网络的运营效率，降低运营成本。在青龙系统中，提供车辆全生命周期管理，包括车辆基本信息管理、车辆维修保养、车辆安全、车辆年审保险等车务管理以及车辆配件管理等。

该系统通过系统辅助调度手段，提升车辆运行效率，降低空载率，提高货物运达的及时率，提升“211 服务”水平。同时通过系统化作业管理，提升车辆作业的标准化，为车辆作业提供监控数据和绩效考核数据。

## 自提柜系统

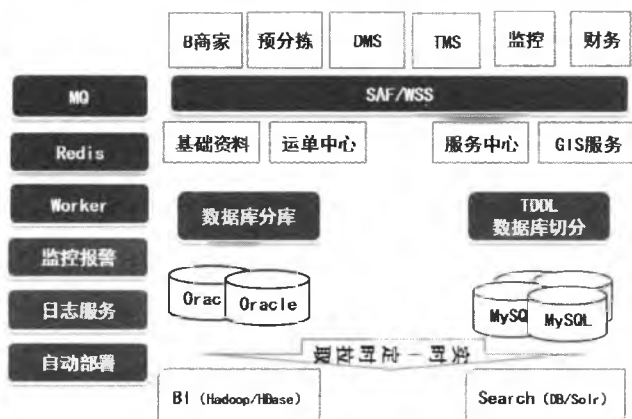
京东自提柜作为物流配送终端，可以解决“最后一公里”的难题，提高配送服务效率和客户满意度，并且可以收集物流配送的大数据。



京东自主研发自提柜

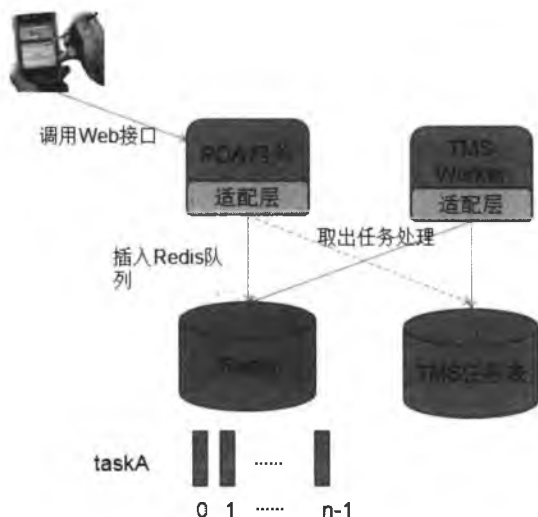
青龙系统快速发展的同时，京东技术也在大踏步前进，青龙团队和公司基础技术团队紧密合作，对系统持续进行技术优化，包括 SOA 框架、分布式调度、Redis、MQ、分布式 MySQL 等，有力地保障了系统稳定运行，提升了系统效率，同时，促进了公司基础技术的成熟。





青龙 2.0 技术架构图

成功在于细节，针对青龙的具体情况，研发团队对技术架构进行了诸多改进，取得了良好效果，例如基于 Redis 的分布式调度，就做到了兼容已有数据库方案，实现了平滑升级、重启等不会丢失数据、高并发、支持批量处理、支持防重注、支持 Redis 故障（自动和手动）切换。



青龙分布式调度架构图

在青龙 2.0 进行期间，也就是 2013 年 618 过后，刘总带来公司的高管到配送站进行体验，引出了青龙系统发展的另外一条发展主线——客户体验。高管体验

完后，对系统客户体验并不满意，提出了二十多个优化意见。因此，我们立了一个优先级非常高的项目，就是高管体验优化项目，并且，把客户体验项目作为一个持续改进的项目，定期派产品和研发人员进入配送站进行实习，不断对客户体验进行改进。经过一年的努力，在 2014 年 618 期间，刘总去配送站体验，对系统给予了高度评价。



刘总 618 配送站体验

青龙研发团队又经过近一年的拼搏，不仅打造了更加完善的物流配送系统，满足了业务快速发展的需求，而且在众多技术取得突破，包括基于缓存的分布式调度、多级缓存系统等，共获得 40 余项技术专利，团队也再次荣获公司级优秀团队称号。

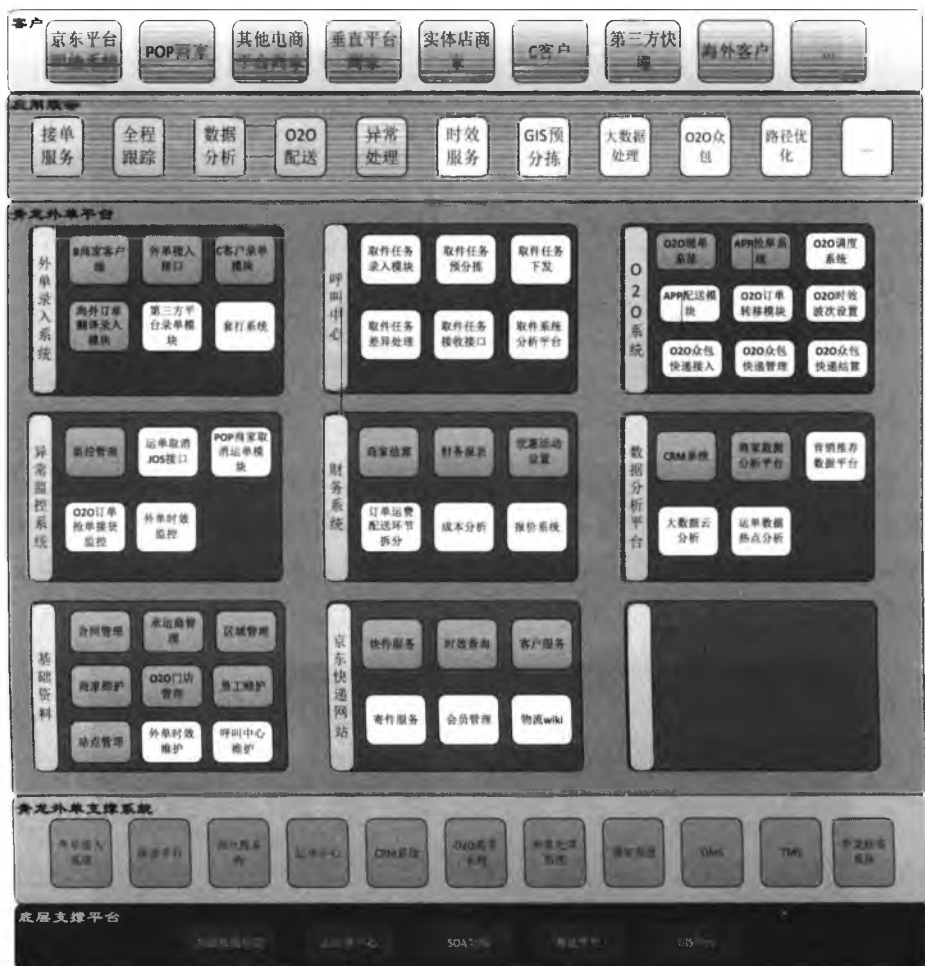


青龙优秀团队合影

## 青龙系统 3.0

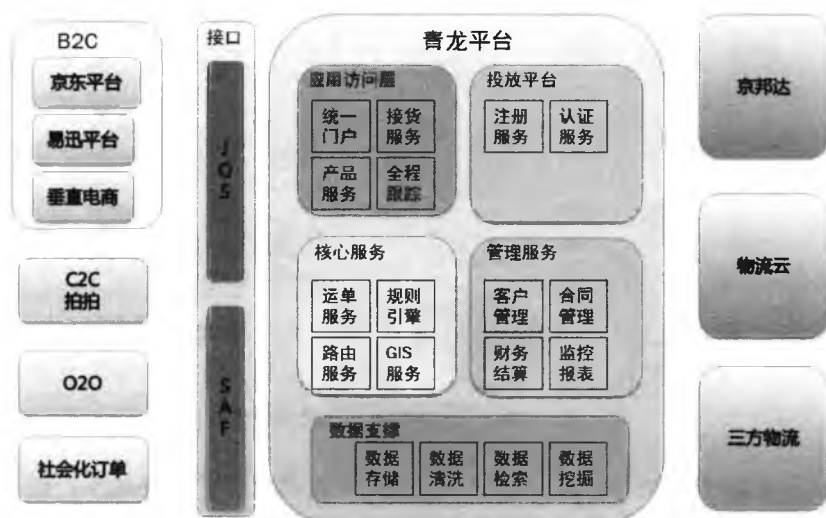
青龙系统设计之初，就考虑到对外开放，并设计了对外开放的子系统——商家系统，但是，1.0 和 2.0 都注重于物流配送本身系统功能的完善和优化，并没有将对外开发置于优先发展方向。

随着京东业务的发展，对物流配送服务开放的要求越来越强烈，特别是随着菜鸟网络成立，配送业务管理方将对外开放放到了最重要的位置，因此，青龙系统也快马加鞭，开始了系统开放历程，完成了 SOP 订单对接和 ISV 对接等重要项目，外单开放平台也浮出水面。



青龙外单模块图

2014 年青龙系统 3.0 自然把“对外开放, 构建生态系统”作为最重要的战略, 特别是随着京东合并易迅物流、收购拍拍等, 整个模式发生了重大变化。



青龙 3.0 模式图

青龙业务模式也开始从京东内部物流系统转变为社会化物流。



青龙 3.0 业务模式

在技术层面, 随着系统对稳定性, 性能和客户体验等要求不断提高, 研发团队在系统分层、跨机房部署等方面也取得了突破。



配送员 POS 一体机 UED 交互提升样例

在团队的奋力拼搏下，青龙系统 3.0 对外开放也取得了重要战果，在 618 期间外单数创出历史新高，开放成果获得公司充分肯定。

### 青龙系统总结和展望

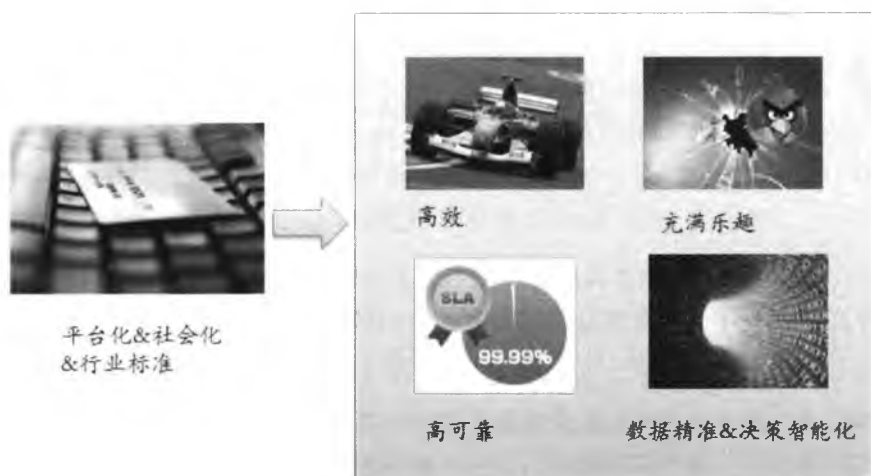
随着 O2O、LBS、可穿戴设备、移动 App、自动化设备、航空运输等业务的快速兴起和技术的高速发展，对于物流配送也提出越来越高的要求。



配送业务和技术展望

青龙系统从满足核心功能的 1.0 开始，到基本功能完善的 2.0，再到 3.0 走向开放构建生态系统，青龙系统从京东内部使用的物流配送系统，向着开放平台化

方向迅猛发展，从而成为社会化物流的重要组成部分，并开始构建起行业标准，那个高效、高可靠、智能化，并且充满乐趣的青龙正向我们走来。



青龙系统展望图

（作者/陈鲲、李鹏涛、吴新红、王茹等）

# 9

## 夺宝奇兵——售后系统

### 追赶业务

2009 年和 2010 年是电商发展最迅速的两年，那时京东被称为“草莽京东”，业务发展飞快。我刚加入售后研发团队的时候，系统可以用简单甚至简陋来形容，那时候很多流程都是没有系统支撑的，每天几千单的售后服务请求，几乎是靠客服 MM 们人肉扛起来的。还记得第一次到备件库一线调研时的场景：库房里堆满了货，货架根本不够用，年轻的京东人穿梭在脏乱的库房内，好多操作员都在用 Excel 记录业务单据。不光是备件库，售后客服人员也几乎在接电话的同时打开着一个 Excel，边处理客户问题，边记录在自己的表格里。每天几千的单量，这种“肉搏战术”可能还能扛，但跟订单量一样，电话、售后服务请求量也在近乎几何倍数增长。John 找到我们研发组，说了一句话“干掉 Excel，全部信息化，3 个月内系统要追赶上业务”。

当时整个售后研发组其实只有 4 个人，我刚加入两个月，但很快也成为老员工了，因为另外两个开发人员也去了别的城市发展，压力可想而知。于是，John 才从其他团队抽调了几名 WMS 高手。那时京东的系统都是 .Net 平台的，老李当时就预见到了向开源平台转型的必要性，所以，我们将要建设的 CRM、备件库生产系统也要求用 Java 平台。

人员不足，业务复杂，开发语言转型，工期紧张，便是当时的背景。在那个窗口期，没有那么多时间让我们犹豫，但研发项目组里的年轻面孔如同当时的京东一样，没有畏惧过，大家激情高涨。

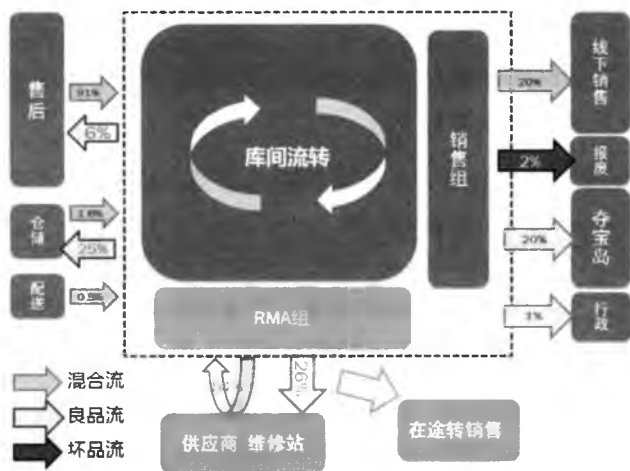
CRM、备件库生产系统两大业务系统同时开工，被列入当年研发部的 10 大项目。老贾，有多年的 WMS 开发经验，和 Liyao 等 8 位大牛负责“备件库生产系统”项目。Yangshuai 和我两个 86 后新人带领一支混编部队（由各兄弟部门支援的、新入职的、.Net 转 Java 的同事组成）负责 CRM 项目。现在想想，那时我们没有太多经验，有的就是老刘输送给所有京东人的“战斗”精神。

虽然面临方方面面的困难，但我们并没有放松对系统建设的要求。一开始，我们就定位要建设标准化流程，辅助业务实现精细化管理的目标。看着简单的一个目标，但在项目刚开始就遇到了难题。

备件库是负责京东售后体系逆向回收商品的管理仓库，当年是京东这辆快车漏油漏的最快的部门。由于对旧货管理不到位，仓库损耗严重，每年损耗在 1 个亿左右。备件库的业务总监亲自加入项目组，与研发团队一起梳理业务需求。不梳理不知道，原来自营电商的备件管理系统是如此庞杂，流程多而细，有别于大库的 SKU 管理模式，退回来的商品都是一单一品管理，尤其是涉及 RMA 厂商送修模块，送修给供应商的货，返回形式多种多样，匹配关系更是难以准确定位。更严重的是，库存周期比较长，光出入库类型就有十多种，然而最叫人头疼的是现有数据如何导入，如何切换上线等等，系统建模难度极大。

这对于一个没有类似库房系统建模经验的团队来说，简直是一团乱麻，无从下手啊。Liyao 主张将主附业务剥离，生产系统与外围审批、监控分离。架构上采用 B/S + C/S 模式，核心业务逻辑都放在服务端，用 Java 平台开发，库内作业系统采用 C/S，客户端用 C#做 WinForm，监控、报表、审批流用 B/S 模式，方便集团管理者操作。库内生产作业又划分为库存、出入库、货位、打包发货、盘点、RMA 等模块。这样做有几个好处：第一是业务核心逻辑封装，复用性增强，便于维护升级；第二是生产与管理互不影响，优先保证生产的正常进行；第三是将外围系统的对接做适配处理，逐个替换，这样就有效地控制了系统改造范围，也便于和现有遗留系统、遗留数据对接，保证了系统平滑切换上线；第四是人员上可以用其所长，.Net 程序员擅长客户端的开发，WMS 过来的高手对于库内管理比较擅长。





备件库生产系统主要业务流关系

很多难题都是大家一起“头脑风暴”拿下的，那个阶段确实是“痛并快乐着”，前期每天都会遇到各式各样的问题，挑战着团队，但看着那些难题一个个被妥善解决，又是那么的有成就感。2010年上半年，项目提前上线发布，备件管理水平大幅提升，当年同比止损八千多万，业务方当年也是满誉而归，而我们售后研发部也收获了一支“骁勇善战”的研发团队。

视角再切回“CRM 战场”。Yangshuai 负责产品设计，我负责研发，在此之前对呼叫中心系统都没有太多了解，摆在我们面前的问题有两个：第一是集成电话系统，预判呼入事件，呼入事件升级转移。第二是以事件为线索，将现有的客服用到的子系统串联在一起，形成一个客服人员的工作平台。相比备件库系统而言，这一版本的 CRM 系统流程相对简单，难点在于要对接整合的外围系统非常多，系统响应及时性也要求较高。

我和架构师 Lidong 带领团队完成了对系统的分析设计，采用以 Case 为核心的模式，将各种客户诉求转化为 Case，客服人员对 Case 跟踪负责，每个 Case 又可以升级、转移。Case 也作为外围系统的串联线索，将“售后服务”、“赔付”、“催单”、“工单”、“投诉”、“延保”、“订单管理”等多个系统当作 Case 的子任务，各外围系统与 CRM 通过消息队列、调度任务来交互，这样的架构做到了解耦的目的。CRM 系统还有个特殊的背景，就是部署在宿迁呼叫中心，京东其他 ERP 系统都是在北京机房统一部署，这中间就涉及一个链路打通问题，Lidong 将系统依赖关系设计为 CRM 单向依赖其他系统，保证链路单向代理即可。

这里边还有个插曲。为了集成电话系统，我们带着测试 MM、开发人员直

接杀到了电话系统供应商的办公室，现场跟他们联调了好几天，别人下班了，我们还赖着不走，直到系统联调通过，第一次让外界同行领略到了京东研发人的风采。

Yangshuai 则负责到各个系统负责人哪里协调接口，大小系统共计十多个，涉及接口则更多，要协调好几个团队资源来支持，本以为这块儿的风险很大，没想到各兄弟部门都很快给出了响应，真心点赞。系统如期上线，但我们到宿迁现场推广第一天就受挫了，来电弹屏功能要好几秒才能打开，看着客服 MM 在接电话时一边跟客户交谈，一边干等着系统的白页，在旁边观察的我们也急得一头汗。数据库依赖太重了，事务里过多地依赖了远程调用。那一夜团队成员都没睡好，大家都在想着各种解决办法。分拆、缓存、异步化消息机制、SQL 优化等等，给系统做了很多手术，好多宝贵经验都是在那个时期积攒下来的。当年系统同时在线用户一千多，每日主事件量 4 万多，平稳的度过了第一个 618 的冲击，当年 618 单日事件量达到了 8 万多，现在的数据是，每日支持事件量 16 万多，618 当天 25 万多。在感叹京东业务快速发展的同时，也只有我们团队知道这过程中的种种心酸和快乐。

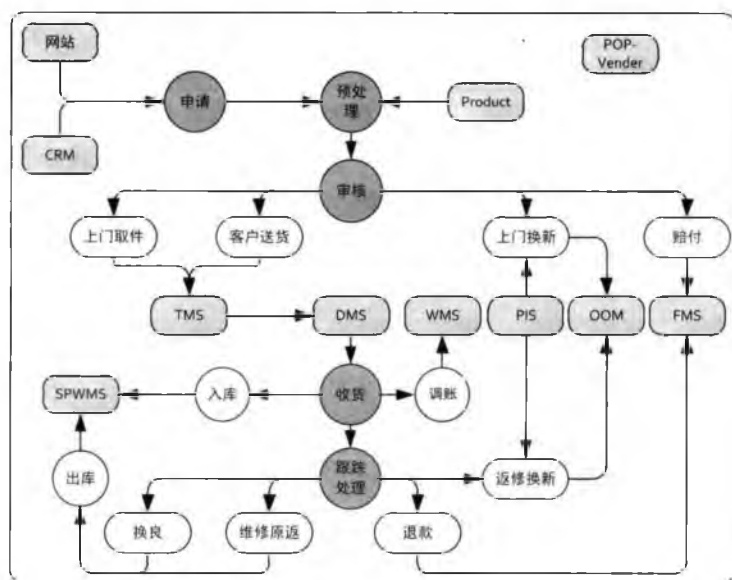


折磨了我们许久的弹屏页面

2012 年年初，备件库、CRM 系统已经比较完善，子系统、模块就有十多个。另外一个核心系统——售后系统，修修补补，一直处于业务挑战技术的状态。老谢，我们的部门经理，敏锐地意识到售后系统作为售后客服体系的核心业务系统，其

发展趋势相当迅猛，几乎是京东逆向流程的总入口，营销系统的业务增长都会影响到售后系统，现有的售后系统架构已经远达不到未来业务的发展要求。又一场恶战拉开了帷幕，集结了 16 人的研发组，耗时两个半月，全新打造售后系统，并正式升级更名为“售后服务平台”。

这时的团队阵容相比两年前堪称豪华，成员里有一半都是“身经百战”的老兵了，但是挑战仍然是史无前例的。前边说了，售后系统是京东逆向流程的总入口，与营销交易系统遥相呼应，也是京东订单生产联调的末端，几乎要和正向所有核心系统发生交互。单纯一个退款子系统就依赖了六十多个外部服务。一方面是京东这种数据岛式的小团队模式造成的，另一方面也体现出了各系统狂野生长的状态，各系统之间都是 SOA 化服务交互。我们的售后服务平台系统，创造了一个记录，就是和外系统交互接口最多的系统之一。



售后服务平台与主要系统的关系图

业务上，流程复杂度也不亚于其他任何系统。大体上我们把系统规划成两大部分：自营体系售后系统和自主售后系统。自营售后系统占总业务量的 60%，涵盖了“退换货”、“维修服务”、“大家电安装维修服务”、“虚拟商品售后”等；自主售后则主要指的是让商家、厂商接入系统直接服务客户，包括“商家（卖家）自主售后系统”、“厂商（直销）自主售后”、“O2O 售后”等。

到 2012 年年底，京东售后客服系统的三驾马车已经打造完毕，即 CRM、售

后服务平台、备件库管理系统。售后客服研发部也从最初的 4 人组成长为 40 人的研发团队。

## 客户体验至上

正如老李说的研发三步走：“支持业务”、“驱动业务”、“引领业务”，我们用了两年多的时间来追赶业务，一路小跑，没敢有丝毫懈怠，因为我们从未忘了我们的最高使命是驱动、是引领。2012 年年底，售后客服体系提出了“简单、专业、快”的要求，大力提升客户满意度，这与我们系统的发展规划不谋而合，要想简单、专业、快，就离不开系统。

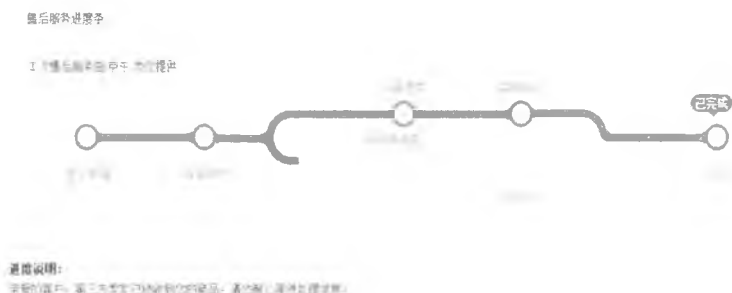
对于我们售后客服研发部来说，我们的客户分为内部客户、终端客户（京东会员）、商家供应商以及兄弟部门。2012 年年底，为了让业务快起来，让客户少等待一点，我们的系统产品做了大量自动化升级。

对于终端客户，也就是京东会员，围绕着客户需求来思考，其实客户满意不满意有几方面的重要因素：

1. 客服的响应速度、处理进度要让客户能感知到，服务流程透明化；
2. 沟通反馈渠道要广、要方便及时；
3. 让客户有更多的自主权，更人性化的服务。

抓住了以上几个痛点，我们就有针对性的用技术手段做了很多工作。比如，最典型的是我们开辟了移动端的服务入口，打通了在线 IM。其中最典型的事件，其实是一个很小的细节调整。不知道使用京东售后服务的你们，有没有注意到在主站 PC 端“我的京东→返修/退换货”频道里，有个叫做“进度条”的东西。这个进度条也可谓是京东乃至整个电商行业里最复杂多变的进度展示工具了：您的售后服务申请单，在什么情况下被处理成什么结果，下一步将会在什么时间响应等等，一条小小的进度条组合出来了将近 20 种展示情况，让客户可以一目了然的知道自己的服务请求处理进度。别小看了这个功能，也凝聚了研发人员不少心

血，将十多种流程的不同分支都用配置的形式来实现，这也是个创新。



多变的进度条

让客户有更多的自主权，享受人性化服务我们逐步改进系统功能。比如，你提交了服务申请单要换货，但是这周要出差，下周才方便取货，以前的做法是让客户下周再提交申请，可是下周提的时候可能就超过了质保时效了，后来我们加了“预约取件”功能，客户问题迎刃而解，配送人员的取件成功率也大大提升。类似的系统改进还有很多，比如用户现在也可以预约客服回电，让客服主动外呼，再也不用为苦苦打不进客服电话发愁，看似很小的一步，对于客户体验提升来说都是很大的一步。

对于内部客户，即售后、客服一线人员，Yangshuai 等一干产品经理可谓费尽心思，他们跑到一线，调研系统的每个操作环节，拿回来了大量简化操作、优化流程的方案。备件库做了采销预处理模块，让库内流转更快；CRM 的工单子系统则打破了各业务部门的沟通障碍，让跨部门的协同服务都能在系统里及时响应；售后服务平台，在原有的“上门换新”特色业务基础上，增加了“换新、退款自动审核”、“闪电退款”、“收货前置”等等。这里边有个很酷的名字“闪电退款”——我们想让客户在发生退货行为时可以快速拿到钱，那从哪个环节来压缩客户等待时长呢，我们的各种自动化处理已经把审单等环节压缩到了分钟级别，最后我们瞄准了返件收货这个环节，可不可以改变传统观念，货物没回收时我就给客户先退款呢？这是个在传统行业看来近乎疯狂到不可理喻的做法，但我们实现了！

那么，这中间的贷款风险怎么把控呢？这要感谢我们的大数据研发同事。大数据时代，我们可以拿到客户在京东的所有行为数据，提炼挖掘这些数据，可以得出客户的信誉度模型。有了这个模型，我们系统就可以智能地识别出哪些是高信誉的客户，对于这一类优质客户，我们就可以开放先退款后取货的业务。



说起数据的威力，有一个很典型的场景，打过客服电话的很多会员朋友或许都会有类似感叹。京东主站上有个会员肖像描述的功能，其实我们的 CRM 系统里也有类似功能。一接到电话，客服 MM 可以主动叫出您的名字，主动询问您是否是有关某商品的某类型咨询问题，以及您的所有信用、偏好，你甚至有时候，都会觉得客服亲切的就像你的邻居，非常有默契。这也依赖与大数据的挖掘，在电话这边的客服通过 CRM 里的描述信息，可以对您的问题作出预判。这样做，跟客户的沟通更有效，更能拉近与客户的距离，提升客户的满意度。

从自动化到智能化，售后客服 3 大系统一路发展而来，随之我们售后客服研发部也有了自动化、智能化。开发组件、框架的积累，让系统开发少了重复劳动，自动化测试平台 AMP，更是解决了复杂流程回归测试难的问题，这些自动化工具也是这么多年来系统无事故无重大 Bug 的有力保证。

## 融合开放

2014 年，对于售后客服系统来说，还是不平凡的一年，这一年公司与易迅、拍拍网融合。从宣布融合消息到系统对接上线用了不到两个月。这不光是对接难度上的挑战，更是全新业务模式的挑战。售后、CRM 系统要开放给易迅、拍拍网，支持不同营销平台的服务请求。

在此之前，售后客服系统也在探索着各种开放模式。比如备件库系统从 2012 年开始就与供应商 VC 平台做对接，将 RMA 流程完全信息化；开放承运商发运系统，方便与承运商的对接；上线竞销平台，让竞购商可以在线竞购备件；售后系统对接 JOS，开放 API，让商家可以将京东售后集成到自己的 ERP 系统。概括的讲，这个阶段的开放模式分为两种：一是直接做系统放在公网，让外部客户访问。二是开放 API，ISV 帮助整合服务。

哦，忘了介绍了，其实还有个开放，就是对内部员工的开放——内销平台。这个系统上线后，据说每天上午 10:30 大家都不敢轻易去上厕所，稍不留神，心爱的 5 折宝贝可就抢不到了，这是老刘特批的给员工的福利，可以 5 折购买备件

库商品，当然了次数有限制。这个系统上线在内部也引起不小轰动，相当于做了个内部秒杀系统。开发内销系统的同学一度成为万千研发同学争相交朋友的对象，大家都在关心是否有“内部后门”，当然是玩笑，不过我们团队的影响力也是在那个时候一下扩大起来了。

易迅拍拍模式则让我们更加看清了系统全面开放时代的来临。从技术上来讲，系统需要剥离出核心 API，用于服务内外部系统、平台的对接，或者将系统打造成可以复用销售的产品；从产品业务规划上讲，我们要开放的是京东的服务能力，这个服务能力包括计算资源，也包括整个客服、售后服务体系资源。

事实上，我们的系统已经走上了开放化转型之路。2014 年，Chenwenfeng 带领 CRM 研发团队全新升级了 CRM2.0，支持多个业务单元的接入。通俗的来讲，就是除了京东的业务，其他商家、厂商、电商平台都可以接入我们的系统，而这些只需要做简单的定制化配置就可以实现。目前 CRM2.0 经历了半年的研发后已经上线推广。



CRM2.0



## 新模式

如果说 CRM2.0 是开放系统的新模式实践，维修中心系统则是业务能力开放的探索。

2014 年上半年 3 月 15 日，京东售后新闻发布会，“售后到家”闪亮登场，而就在这个时间点之前的一个月，我们团队的工程师还在通宵达旦的为该服务上线做准备。

您的一个电话、一个在线申请，京东维修工程师到您家，让您足不出户就可以享受到服务。现在的宅男宅女们还是比较多的，这项服务也正是看中了这点。马上，会员朋友们就不用因为自己的商品（不管是否在京东购买）出现问题而东奔西走，还要提防不良维修商的漫天要价了，可以通过我们的系统在线购买维修服务，全流程公开透明报价。

这是一次尝试，也是一个进步，京东的品牌形象在客户心中有较好口碑，而维修服务这一深水区也鲜有电商试水。我们的维修中心系统，目前支持的是保内维修以及授权维修，马上就会有全新的模式推出。届时，服务即产品，会员朋友们可能通过我们的系统来实现更加便捷、透明的维修服务。这也是售后业务从成本中心到利润中心的一个重要转折，我们给系统起了个美丽的名字“展翅”。

从 2009 年到现在，售后客服系统的精彩篇章一篇连着一篇，现在新的一篇正在掀开，就像振翅待飞的蝴蝶，相信会给行业带来更大价值。（作者/张华）

# 10 华尔街——财务系统 演化史

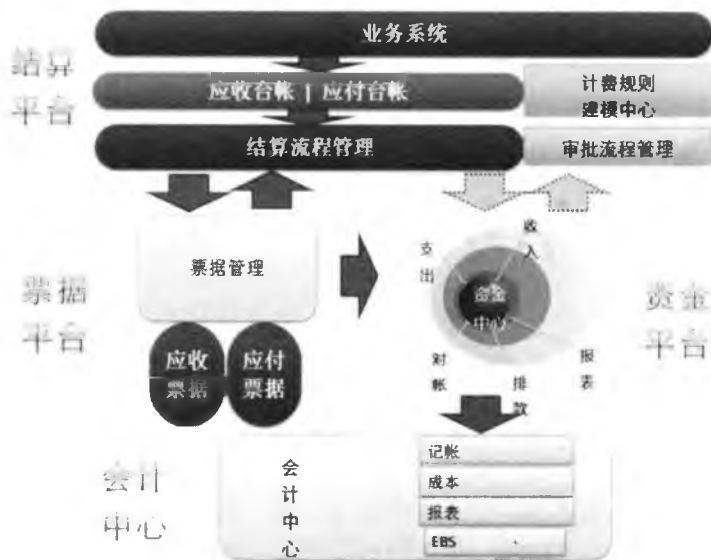
## 京东资金的秘密

京东财务体系包括结算、发票、资金 3 大模块，每个模块又包含着各自不同的功能，如资金还包含订单的正向、逆向的支付、退款流程等。下面，我们来了解一下鲜为人知的京东财务。

资金系统的进化史分为下面 3 个阶段。

### 第一阶段：支付手段单一、资金分散、订单手工对账

这个阶段，由于电子商务刚刚起步，整个电商界支付手段都比较单一，在线支付只能使用网银，货到付款也仅支持现金，这些都是客户能感知到的。客户所不知道的是，这时的京东订单还使用手工对账，当客户支付款项到与订单机构不一致的收款账户时，收款公司出纳和订单所属公司出纳需要进行沟通并做资金调配处理。



京东财务体系

## 第二阶段：支付方式增加、资金集中、系统对账

业务刚刚开展时，各分公司对自己的订单进行收款、对账的方式完全能满足公司的业务需求，但是随着京东市场份额的增加，订单开始呈现井喷式增长，各公司用于订单运营的成本、跨机构订单及后续资金订单收款划拨的沟通成本都在不断增加。因此，财务部门和职能研发部门下定决心要对这种各自作战的方式进行彻底改变。

在会议室里，该业务的产品经理德哥和研发工程师老陈、老牛、老谢几个人都没有说话。其实大家都知道当前问题产生的原因，也对系统进行了多次升级，但是并没有从根本上解决问题。

业务方娟姐不愿让这次会议又以系统升级结束，于是她扔出了一磅重弹——财务部门愿意整合订单收款，由目前的各分公司收款改为由集团统一收款、对账，资金统一管理。这时产品经理和几个研发工程师都是眼前一亮，如果集团统一管理资金，那就可以重新设计系统，除了满足业务的需求，还可以结合自身对京东业务的了解以及对未来的展望，规划一个更高效、完善的系统，规划蓝图，这才是他们加入京东的目的。有了业务方的承诺，几个人争相发言，对系统的规划、对业务的改进……颇有些指点江山的气势。

但是现实的情况是，财务部已经建立宿迁订单中心，用于集团统一管理订单，

可是系统在哪？总不能让宿迁的同事手工处理订单，研发却花两年时间实现这个完美的设计吧！在现实面前，没有人能不妥协。2010年，统一结算一期项目在这样的背景下立项了，放弃了完美的规划，基于现实情况，确认核心和紧急功能，几个人用最短的时间完成了最基础的功能，使宿迁订单中心可以正常运转。

同年，为了满足业务方的需求，统一结算开始了二期、三期项目。通过三期的建设，系统已经可以基本满足业务需求，保证订单的流转、财务的结算，完成规划、建设财务的核心系统的工作，虽然历经方案的各种调整、变更，甚至多次推翻重新设计，但是系统最终成功上线并受到业务方的好评。

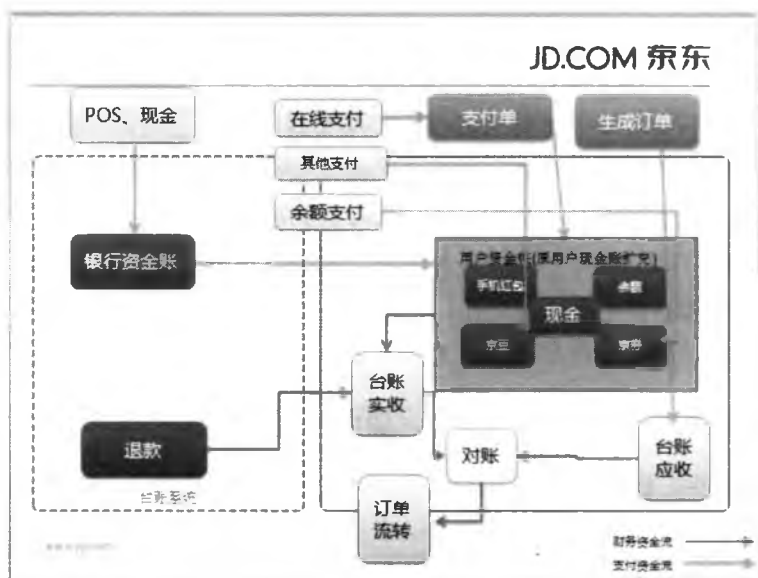
### 第三阶段：支付手段多样化，资金统一管理

在2012年，由于业务的大量开展，当时的统一结算再一次受到了挑战，仅仅实现订单的收款、对账功能已经不能满足财务的管理、操作需求，也不能支持客户日益多样化的付款需求，比如，合并付款、分次付款、混合付款等。新的变革需求摆在我们面前。

这一次业务方和研发部都派出了团队精英，双方配合调研系统现状，分析问题，梳理流程，从业务和系统两个层面分析问题。研发部派出4名产品经理德哥、小谢、小刘、小谭，业务方仍然是娟姐带队，并从宿迁调来订单中心小尚和资金管理中心小爽。3个业务人员针对目前情况，整理问题、梳理业务流程。4个产品经理分工，按照业务类型进行调研，从订单下单到资金归集每个步骤都进行分析，调研范围涉及跟京东相关的三十多个系统，四十多种业务类型。4人白天调研，晚上整理调研文档、交流经验、分析问题、讨论解决方案，历时两个月终于完成了系统现状的调研。经过认真地产品调研，针对当时存在的问题，以及对未来的规划，职能研发部给出了详细、完善的产品设计。

给出方案后，业务方和产品、研发相关人员进行了封闭评审，娟姐、小尚、小爽提出了很多特殊业务类型以及未来可能用到的业务，研发部的4名产品经理根据设计模拟各种系统流转情况，研发考虑系统实现。随着一个个业务场景在模拟场景中的正常运转，几个人紧绷的脸也渐渐有了笑容。在这种精益求精的产品设计理念下，统一结算四期不仅满足了客户支付的需求，还建立了京东资金的账务体系，为后续资金管理、监控建立了最坚实的数据基础。

产品设计完成，只能说系统实现刚刚迈出了第一步，后续的研发和系统切换才是重中之重。由于涉及系统多、范围广，系统的协调以及切换历时半年，也是这半年的切换期，让系统发现问题、解决问题，保证了系统后期的稳定运行和超高的兼容性。



支付流程图

统一结算四期可以快速支持客户的各种支付需求，实现了支付方式和订单对账的解耦。在资金流重新规划后，客户可以更清晰地了解自己在京东的资金使用情况，增强了客户的归属感。银行资金账的建立，结束了京东长达 8 年的资金各自为战的情况，改变了通过单据统计资金的情况，真正实现了通过一本账监控京东集团所有资金流动的情况。在银行资金账的基础上，京东集团实现了订单收款资金的分配和资金划拨。在银企直连的支持下，通过银行资金台账，京东集团可以每天自动生成银行余额调节表，更加大了对资金的管控力度。银行资金账也成为京东财务系统中最核心、最准确的一本账。

## 结算——京东和供应商不得不说的故事

京东给供应商的结算，在 2011 年 1 月之前是通过半手工半系统来实现的。当采购人员需要给商家结算时，结算中心会提交一个手工填写的纸质单据到财务，

出纳根据纸质单据上的入库单号逐一在系统中点击结算，全部成功后，这个单据就结算成功了。而对于结算中心，一个结算员负责管理几家到几十家的供应商，全由纸质单据来管理结算历史非常累。

在 2011 年年会上，财务要求必须使用线上结算。实现供应商线上结算的系统在 2009 年中旬就开始开发了，但由于对业务的不了解和沟通不到位等原因，系统一直没有上线，仍处于产品规划和研发的测试阶段。在 2011 年 1 月 10 日这版开发了一年半之久的供应商结算系统终于上线了。但由于问题太多，上线不久后就被迫下线了。对系统各个功能进行梳理，去掉了所有功能定位不清楚的台账，改良后的供应商结算系统在 2011 年 1 月 20 号重新上线，虽然问题还是有一些，但已经不影响基本功能的使用了。在使用过程中不断收集和整理问题，并于 2011 年 10 月重新设计流程和规划了一个系统升级版，该版本修复了使用过程中的各种 Bug。

供应商结算系统是财务部门第一个实现在线管理对外结算的系统。之后如雨后春笋一般，电子书结算、延保结算、POP 结算等系统相继出现。随之而来的是新的问题：结算系统太多导致研发的维护成本很高；同一家供应商可能会和京东的不同结算系统打交道，一是让商家觉得京东内部管理制度混乱，二是如果商家在京东的一种结算业务中发生违约其他业务系统不会知道；对于京东的应付款项，财务无法看到一个完整的数据。所以在 2013 年研发部规划了结算平台项目以解决这些问题。

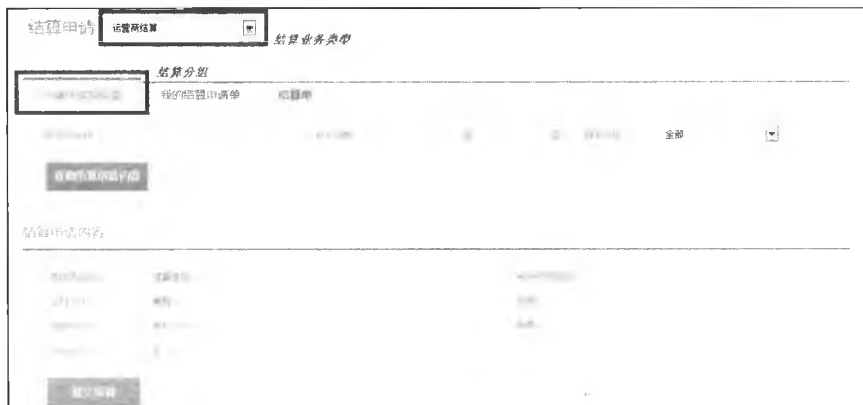
结算平台怎么做？这是财务研发团队设计方案时考虑的最关键问题。项目组从既有系统中分析优缺点、从套装系统中挖掘成熟经验。方案确定的过程非常艰难，往往是一个思路刚刚提出，立即就有反驳的声音出现。计费功能要不要、台账怎么做、页面用什么风格、业务流程怎么优化等太多的问题需要考虑，太多的因素需要平衡，直到研发部的牛架构同学敲定结算平台的业务架构和设计架构所有人的思路才统一起来。领导一声令下，开干！

有了定海神针，产品、研发、测试小组迅速制订项目计划，这第一步稳稳地迈开，项目组历时 3 个月的通关过程也就此开始。

## 第一关：产品关

产品设计是起点，这么“高大上”的系统要是没有好的产品设计，如何给研发兄弟做榜样。但是项目工期实在是太紧张了，留给产品设计的时间只有两周，没有时间去精雕细琢，怎么办？7 个产品经理全力以赴画原型，分模块并行工作。原型出来后，讨论、修改、再讨论、再修改，期间三易原型，总算把页面、功能、

产品说明统统搞定。在研发工作启动后，产品经理邀请 UED 小芳同学对原型进行再设计。半个月后，一个风格清新的产品原型摆在了产品经理们面前，大家的第一感觉是：这就是我们想要的。下图为证：



结算申请产品原型



待办事项产品原型

## 第二关：开发关

经过产品部门同事的不懈努力，一个功能强大、流程清晰、页面风格独特的系统展现在我们面前，但该系统业务复杂、功能繁多。面对这么庞大复杂的项目，我们如何快速细致地理解系统逻辑结构？如何迅速完成开发文档？如何在有限的项目期限内保质保量地完成项目？

### 1. 理解和熟悉结算平台流程与功能。

由于时间紧迫，架构师只能完成基本的开发设计文档，但系统设计涵盖的业务众多、逻辑与功能复杂，开发人员通过简单的项目文档和基本的开发文档无法快速地理解系统业务与结构，我们采用不断讲解、不断细化的方式让开发人员快速地熟悉系统。在开发人员对系统有了初步了解的基础上，小组一次次地组织会议沟通，由架构师讲解，开发人员消化，不懂就问，直到开发人员能够完全理解整个系统。

### 2. 项目开发分配。

由于系统复杂、工期紧、开发资源紧张，领导决定商家结算组 99%的人参与项目。其中，部分人员全职投入，其余人员兼顾原有系统的日常需求。整个开发小组根据系统模块分成 3 个小组，第一小组负责系统配置、审批流程、接口等模块，第二小组负责结算申请模块，第三小组负责发票模块。

### 3. 一期项目开发与提测。

针对运营商结算，整个开发过程中，我们采用齐头并进、拨点作战的策略，也就是：架构师针对当期功能不断细化开发文档，UED 不断细化完善脚本，开发人员针对各个功能拨点攻破，并且研发团队每天汇报进度、总结开发中的问题、制定第二天的开发目标。整个开发过程高效有序，在与其他部门研发同事的通力合作下，开发人员在规定时间内完成了系统的功能开发、内部功能测试、系统间联调，并最终交到测试人员手中接受进一步的检验。

### 4. 二三期开发与后期展望。

在一期的基础上，我们二三期迅速完成了 POP T+1 结算、校园合作站等系统的接入。对后期结算平台的开发，我们致力于模块细致化，以最小的开发量完成最优质的结算业务。

## 第三关：测试关

测试前期，面对生疏的新业务，测试人员理解起来十分困难与纠结，想不通的逻辑扰乱了他们的思路。初期的需求理解在后续整个项目测试中占了很大的比重，主导整个测试过程的走向，同时，项目测试的成败全在于测试初期的设计及用例的完整性。测试小组不断地讨论测试用例的设计，召开测试用例评审会，以求业务场景全覆盖，测试用例的尽量完整。



在牛架构给研发和测试开的 N 次需求讲解会中，我们逐渐梳理出来了业务场景，并深入掌握了系统的设计及模块功能。完成了测试计划的编写及测试用例初稿。通过架构、研发和测试的共同评审，不断改进、不断完善的测试用例终于完成。

测试过程中，项目经理协助我们与各个部门沟通协调争取测试资源和数据，打通各个上游环节，确保测试期间联调的数据保障及测试资源准确到位。测试小分队分工明确，努力地验证系统的逻辑及可用性，不断地提交系统 Bug。

继第一个运营商结算业务后，系统又逐步接入 POP T+1 结算等多个结算业务，项目的架构逐渐明朗，测试小分队逐渐熟悉了系统的模块设计，工作变得得心应手。在不断地进行迭代测试、验证过程中，项目由混乱变得清晰。测试与开发人员不断地讨论如何更好地完善系统，共同为系统的优化做出努力。

测试接近尾声，项目的缺陷全部被修改，结算平台项目终于新鲜出炉，在大家共同期待下上线了。测试小分队通过供应商结算总结的经验，及结算平台测试中遇到的问题总结出了测试应把握的几点原则，并坚持不懈地遵守着。第一是日清缺陷、延期缺陷实时跟踪；第二是阶段性总结测试数据及结果；第三是初期与产品人员紧密沟通，确保需求理解的正确性。

经过一段时间的试行后，测试的体会主要有两点，第一是团队齐心，其利断金，第二是软件测试的真正意义在于发现错误，而不在于验证软件是正确的。

再严密的测试也不能完全发现软件当中所有的错误，但是测试还是能发现大部分的错误，最起码能确保软件基本是可用的，所以在后续使用的过程中还需要加强快速响应的环节，结合软件测试的理论，使故障暴露在最终客户端发现之前，并及时主动地去解决。

## 结算平台上线

2013 年 9 月，经过财务研发团队的拼搏协作，一个全新的结算系统——结算平台顺利上线了。新平台与以往的结算系统相比，提供了统一结算入口、统一结算流程、统一结算台账，为各结算系统的整合奠定了坚实的系统基础，是京东结算系统发展历程的一个重要里程碑。

### 结算平台架构特点

1. 新建计费规则池，提供计费规则的前台配置和展示功能，为财务对结算计费的风险控制提供便利；

2. 重构应付应收台账，实现与业务系统的解耦，明确结算系统功能边界；
3. 增加结算分组概念、区分结算申请单与结算单功能，满足不同采销业务的结算需求，提升系统的兼容和扩展性；
4. 统一的结算架构，统一的结算与发票管理系统、收付款管理系统的接口，降低结算系统运维的风险和工作量。

### 结算平台整合历程

#### 1. 目前完成的接入

- 2013年9月，运营商结算业务接入。
- 2013年10月，POP T+1结算业务接入。
- 2014年4月，校园合作站及联营点运费结算业务接入。
- 2014年5月，虚拟运营商结算业务接入。

#### 2. 未来计划接入

- 2014年7月，云平台服务市场结算业务接入。
- 2014年8月，海外采购结算业务接入；仓储服务结算业务接入。
- 2014年10月，青龙干线运费结算业务接入。
- 2014年11月，备件库维修结算业务接入。

.....

### 结算平台未来规划

#### 1. 自动化结算

在新业务接入结算平台的过程中，结算平台的功能将逐步优化，朝着自动化结算方向打造更加健壮、灵活的结算系统。

- 个性化结算规则配置——自动提交结算。
- 个性化结算审批流配置——自动结算审批。
- 个性化发票核销规则配置——自动发票核销。

#### 2. VC 结算

依托结算平台，建立供应商结算平台，向 ISV 提供 AIP 接口，打造京东与供应商一体化的供应链结算生态系统。

## 发票——一张纸的前世今生

在国内，发票可以作为收付款凭证、保修凭证，以及税务凭证，因此，京东会针对每一个自营售出的有价商品开具发票，这是对自营商品质量的保证和对用户的承诺。于是发票系统也变成京东众多系统中不可或缺的一个。

### 发票系统的定位

发票系统是一个生成系统，位于系统链的下端，其上游有着订单系统、商品主数据等相关系统。发票系统主要负责整理上游系统的数据，开具出最后的发票，最终交付给用户。



发票系统的定位

### 发票系统的架构



发票系统架构图

经过 2013 年以来发票系统的整体改造，京东已经搭建了发票系统的基础数据层，包含：应开数据、实开数据、开票台账。

1. 应开数据为开发票的参照数据，来源于各个上游系统，包括商品名称、单价、数量、优惠、税率、发票抬头、开票主体等信息；
2. 实开数据为实际开票的信息，主要记录发票票面上的各个信息，如发票代码、发票号码、开票内容等；
3. 开票台账主要用于开票动作的记录，将应开数据与实开数据进行对账，保证应开数据与实开数据的一致性。

发票系统的应用层是基于基础数据层的，它包含三部分内容：发票基础初始化、开票功能、发票服务功能。

发票基础初始化用于发票信息的配置，包含发票模板配置、开票内容配置、开票主体信息配置等。

开票功能针对各项业务需求进行统一管理，例如针对增值税专用发票开具的“增值税集中处理平台”、针对供应商开票的“申请单管理平台”等。

发票服务功能目前包含两个功能：针对增值税专用发票的“资质审核”功能和针对发票邮寄的“发票邮寄管理”功能。

在这些数据和操作的层面之上的就是发票的报表层，用于统计各项发票的数据，如“每日未处理的应开数据”、“每日已开的发票数据”、“每月已开的发票数据”，以及“差异报警”等信息。

## “高大上”的电子发票

我们先从电子发票的业务说起。电子发票的本质是以数据流的形式保存发票的信息，用户可以仅凭借一个 PDF 文档或者一个二维码就可以调用到这些数据，用于日常报销等。而发票也均由系统与系统间的数据对接产生，不再使用实物的购票、打印、报税等。对于京东这种需要高效率处理的电商而言，电子发票可以减少很多人工干预，大幅提高生产时效。不相信的同学可以看看以下两道题目：

题目 1：库房人员打包一个商品需要 3 分钟，打印一张发票并放入商品包装中需要一分钟，这两个工作均完成才算完成一个订单的生产。如果该同事可以不停歇地工作一个小时，请问他可以完成多少订单的生产？如果发票换成电子发票，无须库房生产，请问他在相同的时间内可以完成多少订单的生产？

题目 2：财务退款审核时，金额审核由系统处理，一笔退款金额审核需要一

毫秒的时间，发票由人工处理，一笔发票作废需要 3 分钟的时间。如发票换为电子发票，无须人工处理，请问原处理 10 单退款的时间，现在可以处理多少单的退款？

从电商的运营层面来看，这个项目也为京东节约了大量的成本。假设一张纸质发票的购买价格为 3 角钱，那么，电子发票一年的开具量已为公司节约了几百万的成本。更宏观一些去看，电子发票也为社会节约了大量的资源。我们做过一个数据统计，京东一年开具的电子发票换成纸质发票的话，平铺开约为 5 个北京市的面积。如果将这些发票码放在上海亚洲一号的仓库里，约占据库房 3600 平方米的空间。（作者/刘磊、谭聂、王超等）

# 11

## 成长的烦恼——内部 信息化进程

### 京东特色新员工配型

京东 CTO 条线研发部下属的职能研发部有一个传统，就是部门新员工入职以后，除了要参加公司层面的价值观体系培训、研发部层面的部门专业培训外，还要参加一个名为“老曹和京东的这几年”的培训。

名为培训，其实就是一个自称“老曹”的年轻人和新入职员工的茶话会，主要内容就是听老曹神侃一通他所经历的京东发展历程。这个茶话会既无点心也无茶，我之所以称其为茶话会，点心就是老曹讲的当年跟老刘一起在中关村卖光盘、搞促销时在仓库打包到凌晨 3 点第二天 8 点半准时开早会等等的段子，茶估计就是我听到京东以每年 200% 的增长速度迅速发展、同时这帮创业者个人财富也迅速积累时咽下的羡慕嫉妒恨的口水了。这个老曹，现官居京东集团下属子公司“金融集团”CTO。

我在参加老曹的茶话会时，他说过一句话：“京东是一个互联网公司，同时更是一家贸易公司。”我对这句话一直记忆犹新，同时也越来越深刻地体会到这一点，因为我所在的三级部门“综合职能研发部”所对口的人资、行政部门负责

的内部管理业务在当时的信息化程度与京东“互联网公司”的名头实在是相去甚远。

在很长的一段时间里，京东所有的信息化系统都是围绕着电子商务的核心展开，所以京东的商务、客服、仓储、配送等业务的信息程度很高，而用于内部管理的 OA、流程、HR 管理等则一直停留在比较初级的阶段。好在，经过一段时间集中的整体规划和快速建设，职能管理信息系统已经有了较大程度的提升，其中比较有代表性的是企业内部门户、流程平台两个业务线的产品。

## 京东企业内部门户

话说我来京东之前在一家软件外包公司工作过几年，这家公司专门为集团型企业提供 ERP、SAP 等重量级产品的实施服务，所以在我深深的脑海里，ERP 是一个统管企业的生产、制造、财务、销售、采购、流程等等关键业务的、大而全的系统。

所以，当我顶着“产品经理”的 Title 刚刚办理完入职手续、领导就跟我说“洪涛，以后由你负责京东的 ERP 系统”的时候，我顿时感觉膝盖有些发沉：我大京东果然是疑人不用用人不疑，让一个新员工负责这么重要的系统。后来我知道我想多了，因为京东人所说的 ERP 是专指京东企业内部门户系统。

京东的内部门户系统按时间来分，大概有 3 个阶段，分别是老 ERP 阶段、Portal 门户阶段、一站式办公平台阶段。

## 历史悠久的老 ERP



老 ERP 系统界面

上图，就是我接过京东 ERP 大旗的时候呈现在我面前的系统界面。老刘握拳微笑、刚毅儒雅，以青年企业家的形象出现在我脑海中。

这个系统在 2009 年春天开发上线，一直使用到 2013 年，在京东的信息化系统中也算是历史悠久的了。从 2009 年到现在，京东集团的员工数从 2000 人扩张到了 60000 人，当时参与过研发的老肖、春哥等同学们，后来也都成了京东各个研发条线的 Leader。我们后来在“京东人@work”项目（对 ERP 系统的 Portal 版升级项目）开发过程中，曾经对公司各条线 VP 级以上高管做过需求访谈，其中很多高管都参与过这个系统的研发工作，包括段子手老徐。

老徐说：“这个 ERP 是当年我跟一帮人搞的，我走了两年回来一看，怎么还是它？”这个老徐，现官居京东集团市场营销部高级副总裁，是互联网电商圈内声色俱佳的段子高手。

另外，据我的研发团队的一个兄弟曾跟我说，他在重构 ERP 的一段老代码时发现一段从注释看来是老刘写的代码，未知真假。我问他：“代码水平如何？”他说：“没细看，直接注释掉重写了。”

必须要说明的是，虽然以今天的眼光看，这个系统的界面不是那么炫，但是它在技术架构上有良好的可复用性和拓展性。



在系统的权限控制上，春哥借鉴 RBAC 权限模型自行开发了整个系统的权限体系，这个权限体系奠定了后来的 UIM（统一身份管理）系统的理论基础。现在，UIM 系统管理着京东一千多个内部信息化系统、三千多个资源点，控制着六万多名员工的权限。

另外，经过岁月的打磨，这套系统在功能设计、使用体验上也颇受好评，以至于我们的新 Portal 门户在上线很长一段时间后，仍然有好多人在恋恋不舍地用着老 ERP。

## “京东人@work”

2013 年的 12 月，对于京东集团 CTO 体系“综合职能研发部”部门经理老杨来说，是一段充满了艰辛和欢乐的日子。因为在 2013 年 12 月底，新版门户系统“京东人@work”和流程平台两个重点产品同时成功上线了。



新版 ERP “京东人@work”项目上线后的界面

2013 年 7 月，公司终于决定用一个新的系统来接替伴随了全体员工近 5 年的

老 ERP。这个重担落在了“综合职能研发部”老杨团队的肩上，从此，北京国家会议中心旁边的北辰世纪 A 座二层，又多了几扇灯火通明的窗子。

新门户系统命名为“京东人@work”，寓意是为京东人打造一个一站式的工作平台，项目重点确定了 3 个目标：

1. 协同办公平台，整合所有的内部信息化系统，重点打造“集团门户”、“工作中心”版块，提高协同效率；
2. 技术架构转型，数据库从 SQL Server 转为 MySQL，开发语言从 .Net 改为 Java，系统整体采用 Jetspeed Portal 架构；
3. 统一权限梳理，将现有的三千多个资源点按照业务条线重新分类。

上述目标确定后，老杨亲自担任产品经理，带领团队开始了连续 4 周不分昼夜的产品规划，当产品的交互原型、PRD 全部完成的时候，已经是 2013 年的 9 月底了。

当时定向支持该项目研发的是营销研发部下属沈阳研究院。据说，2013 年 10 月初某一天在沈阳浑南新区昂立信息园 IT 国际二层的会议室里，老杨和研发团队的讨论会从下午 4 点一直持续到晚上 12 点，双方争议的焦点在项目排期。老杨坚持要在 2013 年 12 月底按时上线，而研发团队说按现有工作量估算这个时间无法保证，于是会议的主题就变成了对项目 WBS 中的每一项开发任务的工时讨论。后来，会议室里的唇枪舌剑变成了饭桌上的觥筹交错，老杨众望所归地被一帮东北老爷们儿灌醉了。好在他的酒没有白喝，最终他拿到了研发团队给的上线时间为 2013 年 12 月底的最新 Project。

然而，沈阳研究院研发团队真正辛苦的日子才刚刚开始。

首先遇到的问题是门户前台主体架构的 Jetspeed2 框架。当时京东乃至全中国各大互联网公司尚无使用 Jetspeed2 搭建大型企业门户的项目经验，而且也因如此，在网络上能够搜索到的中文技术文档也寥寥无几，很多帮助文档只能在 Apache 该项目的官网上下载到，但是，鉴于 Jetspeed2 提供的丰富的 Portal 技术以及灵活的 Portlet 管理机制，研发团队最终还是决定挑战一下。沈阳研究院专门选定了 5 个对 Java、Portal 底层架构有丰富经验的架构师对 Jetspeed2 架构进行深入研究，同时又定向招聘了几个在计算机专业英语方面有优势的研究生，支持技术文档的翻译工作。

另一个让项目组头疼的问题是老 ERP 中集成的其他系统的迁移工作。在老 ERP 系统的基本框架搭建完成之后，运营、营销等各业务条线的信息化系统不断地研发和集成进来，加上之前的业务系统，到我们做版本升级的时候，初步估计

已经有一千多个线上系统集成在老 ERP 中，项目组需要考虑如何将这些系统平滑地迁移到新的“京东人@work”系统中。经过多方讨论和权衡，项目组最终采用的方案是“系统单点登录、功能界面嵌入、核心功能集中展示”。研发团队首先针对 ERP 账号开发了统一的单点登录机制，并开放单点验证 API，要求所有线上系统接入单点登录。同时，将各业务系统的功能进行梳理和划分，以便各功能界面嵌入到 ERP 系统中，实现员工一站式的操作体验。另外，这种方案还有一个好处，就是可以通过 UIM 对各功能点进行统一的权限控制。对各业务系统中的核心功能，是通过 Jetspeed 的 Portal 框架提供集中的 Portlet 展现区域。

一个又一个的问题在团队成员的讨论和试验中得到了解决，一个又一个功能也被项目团队在 3 个月内艰辛而又充实地逐渐开发出来。为了保证 2013 年 12 月底能按时上线，项目团队所有成员主动单休，每周只休息一天，在项目的后期，连续 3 周都没有休息一天。过程虽然艰辛，但是这种快节奏的项目过程也培养起了一支优秀的项目团队。直到现在，当时参与过这个项目的几个同事一起聊天时，回忆起当时的情景，总会心有感慨地说：“那真是能力飞速提升的 3 个月啊。”

项目最终在 2013 年 12 月底顺利上线，得益于我们在需求和设计阶段的精雕细琢以及开发过程的努力，系统一上线就收到了良好的反馈。目前，该系统在京东企业内网平稳运行，已经成为每个京东员工日常办公中最常用的系统之一。

## 一站式的协同工作平台

“京东人@work”系统上线，是我们打造一站式协同工作平台的一个开端。从系统上线到今天，我们一直在对系统持续地补充完善，下图就是我们在 2014 年 Q2 刚刚上线的“行政在线服务大厅”模块的界面视觉示意图。



行政在线服务大厅模块界面视觉示意图

在我们的规划里，初步成型的“一站式协同办公平台”应该具备的子系统有：

1. 门户中心，集团、子公司、各业务条线的信息门户集合；
2. 工作中心，企业员工日常工作的一站式窗口，为每个员工汇总出其在各个业务系统中的常用功能，以 portlet 的方式展现，以满足优先级定制化的需求；
3. 任务中心，面向全体员工的、以协同为目的的任务分派、跟踪和执行；
4. 流程平台，企业 BPM 的集中管理；
5. 个人空间，非工作视角的员工个人空间，从而拓展出企业 SNS。



“工作中心”中管理者视角的界面视觉示意图

## 京东业务流程平台

京东现有两个流程管理平台：IT 服务台和流程中心。IT 服务台是基于问题、事件跟踪管理的流程平台。流程中心是 OA 平台中的流程中心，承载着日常信息化的审批流程。下面，我给大家分别介绍一下这两个流程管理平台。

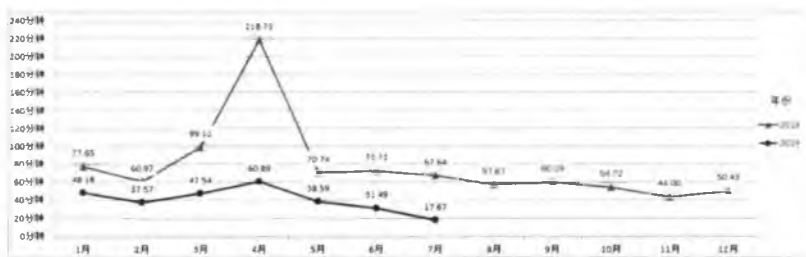
## IT 服务台（ITSM 到 ITSV）

公司发展的速度要求业务流程管理必须做到高效，要想让前台各种业务正常运转最好的方法就是借助有效的流程管理平台。就在此时，公司某部门领导发现了业界大牛 BMC 公司的 ITSM 产品，并以最快最高效的方式于 2012 年年初将其引入京东，并由我们与 IT 运营团队合作共同建立起全公司级的 IT 服务台，统一了 IT 故障/请求处理入口，确立了 IT 系统故障/服务请求的处理规范，在一定程度上提高了 IT 运营/运维的效率。

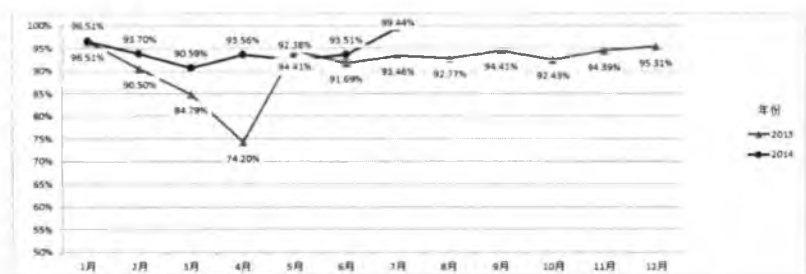
但是，由于该产品诞生于国外，所以存在着各种非本地化操作习惯，导致我们在使用过程中并不是很顺手。为此，我们的研发大牛们灵机一动，想出一个高效的建单方式，让用户以邮件的方式反馈并将其作为问题报告入口，后台支持系统识别后将该问题在 ITSM 平台上自动生成对应的事件单，并流转至下一个操作环节。通过这种简单的操作方式大大提高了事务处理效率，从而，有效地提升了客服、采销、配送等所有业务部门对 IT 系统服务的满意度。

我们的研发团队为更好地解决非本地化的操作软件给我们带来的不适应，从 2014 年以来，一直致力于自主研发基于电商行业背景的 ITIL 落地系统（代号：ITSV）。

以下两幅图充分体现了 IT 服务台给我们带来的价值：



近两年 IT 运营工单平均耗时趋势



近两年 IT 运营工单解决率趋势

## 流程中心

流程中心的由来还得追溯于 2013 年 3 月底的一次公司会议上，有位高层领导指出“我们现在的日常事务流程还在走纸质单，确实有点 out 了，所以我们必须要建立相应的 IT 解决方案，尽早脱离纸质单，走向电子化办公道路”。

会议结束后，领导就带领我们马不停蹄地查找各大 OA 企业办公平台相关的信息，并与其筛选出的几大厂家进行逐一的深度沟通交流，预想从其中选出符合我们京东的 OA 平台。但是，数次与各大厂家的交流都未能让我们找到心满意足的 OA 平台。此时，耳边响起了毛主席的口号“自己动手，丰衣足食”。最终李老大决定，我们大家齐努力来创造京东自己的 OA 平台。

目标确定后，老大“钦点”了几名经验丰富的产品经理对 OA 平台进行全面的系统规划，但由于 OA 平台涉及的范围较广，因此，第一步先着手建设流程中心，尽早实现电子化审批。最终将流程中心产品规划定位为：以快速、灵活、简便为核心理念，实现个性化的表单配置、流程搭建，及邮件审批模板个性化的字段定义等功能。

记得那是 2013 年的 8 月，项目开发工作正式启动，大牛的研发经理带领了一组小弟们一同走进了原 6 层的上海会议室准备迎接战斗。由于该团队都是刚刚融入京东的新鲜血液，所以很多东西都是第一次接触，需要了解的内容很多。在一开始，技术大拿们一起商讨技术框架的搭建，到后来大家按部就班地进行功能模块的开发工作。在此过程中由于任务重、资源不足、工期短等原因各种困难一一暴露，但是大家依然齐肩并战，将问题一一打散，最终流程中心的第一版正式问世。

在那段艰苦岁月中，为了让研发小伙伴们的开发环境优越一些，领导各方协调最终带领大家转战进入宽敞明亮的广州会议室。一直到现在，流程组的小伙伴们依然在那里办公，只不过现在会议室的名字已改为霸气的“华盛顿”（据说原来名为“华盛顿”的会议室可是当年老刘专用的）。

虽说流程中心第一版主体功能已经可以正常使用，但是还存在不少弊端，例如：不支持对外的接口服务、不能进行更多业务个性化的流程搭建配置，且流程搭建工作较为繁琐等。但经过短时间功能优化、产品升级后，现在京东自主研发的流程中心平台已经搭建了近 100 条流程，成功承载着京东的主要业务流程。

回顾流程中心的这些年，不到 10 人的开发团队让产品从无到有，再到支撑起近 100 条不同业务流程的日常使用，到如今承接统一待办，将其他各业务系统的审批集成到该平台，他们一次又一次地提升了大家的日常办公效率。通过这些

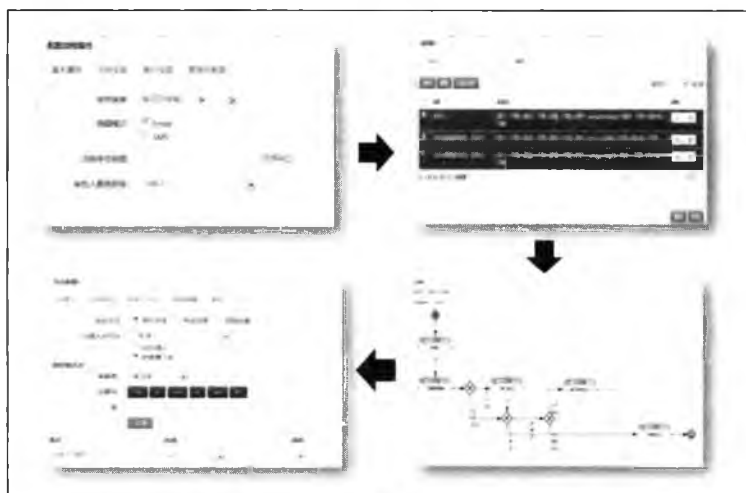
数字及标志性的里程碑已经可以说明流程中心是我们日常办公不可或缺的平台。



流程地图



我的待办



实施方案

(作者/杨洪涛、南婕)

# 技术演进

京东研发是用怎样的技术架构来支撑海量业务的呢？

本篇将通过SOA治理、中间件、云平台等几个最关键的技术架构，为大家展示一个海量处理系统的系统架构，并第一次公开京东是怎么一步步从.Net转换到Java的。

- 从Net到Java
- 中间件
- SOA治理





# 12 少年派的奇幻漂流

## ——从.Net到Java

### ASP 时代的个人网站

不知道还有没有网友记得京东网站最早时期的样子。我曾在 2005 年加入京东之前，就有过一次在京东网上消费的经历，买了两桶三菱 DVD 刻录盘。那时京东还叫作京东多媒体，主营产品是刻录机和光盘。

这一版的京东网站是 2003 年从外部购买的一套 ASP 的在线商城软件。这套软件虽然以现在的眼光看非常简陋，但当时的业务量也不大，已能够支撑当时的业务。当时研发部只有四五个人，老吕、静姐、小毛、平哥，他们在这套软件的基础上逐步维护和升级，推出了优惠券、DIY 装机大师、联盟等系统。

到了 2008 年，这套购买的 ASP 网站系统逐渐显露出其局限性。在访问量和订单量不断增长的情况下，网站经常出现缓慢、无响应的情况。这时，一个重要人物——老李，京东第一个外部聘请的副总裁——出现了。

## 封闭开发升级.Net

2008 年老李加入京东后干的第一件事就是搞封闭开发,把京东系统从较原始的 ASP 升级为.Net 架构。2008 年 7 月 21 日,老李带领着军哥、缙哥、涛哥、春哥、民哥、台哥 6 个研发人员,小宁、大勇两个测试人员,字哥、罗老师、小磊、嵘哥几个产品和 UED,进驻了位于北京通州郊区的一栋别墅,开始了 3 个月激情(wu)燃(tian)烧(ri)的封闭开发。若干年后的现在,上述的几位老前辈基本还都在京东,身处各自重要的岗位。

封闭开发的作息时间是,早上 8 点从床上爬起来开始工作,晚上干到困了为止,一整天除了吃饭上厕所基本都在电脑前写代码,有时候干得兴起甚至会到凌晨两三点,甚至通宵。一周吃喝拉撒全在这栋别墅里,只有周日让大家回家休息一下。

封闭开发的队伍中,最有激情的就是老李本人。他会亲自参与架构设计的工作,甚至动手写部分核心代码。大家晚上工作到两三点去卧室睡觉的时候,看见老李还在电脑前精力十足地敲打着键盘。第二天早上,大家睡眼惺忪地从床上爬起来,惊愕地发现老李依然在电脑前敲打键盘!搞不清楚他到底是起得早还是压根儿没睡!后来,老李的助理光姐秀出来他以前获得全国劳动模范接受温总理接见的照片,大家才恍然大悟——不愧是劳模呀。

那段时间,老刘经常会到封闭开发的别墅看望大家。一是看看大家的成果,另一个就是带大家改善下伙食,偶尔也会亲自下厨,展示下手艺。到了 2008 年 9 月,封闭开发接近尾声,老刘几乎每天晚上都会来别墅看大家的研发成果,并且提出一些改进意见。大家得到意见后,会立即着手改进,尽量在当天就改好,让对用户体验相当挑剔的老刘满意。有一次,春哥兴奋地给老刘演示他完成的一个功能,结果手一抖,竟然点开了桌面的一张惹火的美女图片。老刘拍拍一脸尴尬春哥的肩膀说:“春哥精神头看来不错啊。”

2008 年 10 月,.Net 版的京东上线了。这也是支撑京东商城运行了 3 年的系统基础。虽然很多技术按现在的标准已经过时,但是与最早的一版相比还是有非常大的改善。



2008年京东首页

京东网站在 2008 年封闭开发后的样子，也是最后一版采用蓝色调的首页，2009 年改版成了大家熟悉的喜庆红色。

## 架构升级为 Java、服务化

2010 年的时候，已经开始有一些系统使用 Java 开发，但是核心的商城交易部分依然是以 2008 年那次封闭开发为基础的 .Net 系统。因为从 2008 年那次封闭开发后，交易系统已经过了两年的发展，系统非常庞大，业务分支众多，光子系统已经有三十多个了，升级非常困难。在万般无奈下，我们只能不停地增加硬件来硬扛。但是服务器越多，运维的成本也就越高，上一次线非常困难。那时候还没有自动部署系统，都是让运维的同事手工部署，搞到最后运维同事都躲着交易系统的研发人员。而且这样增加硬件带来的性能提升也是非线性的，加得越多提升越有限，总会有个尽头。

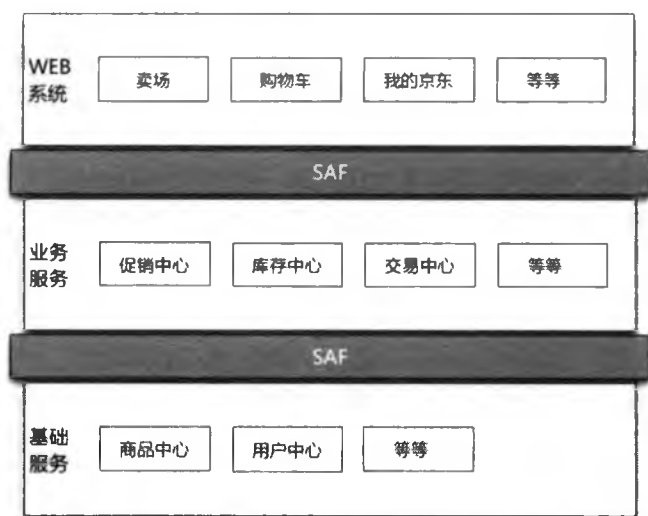
2011 年，有一次图书采销部门搞了一次全场 5 折抢购的活动。促销效果非常好，网友们抢购的热情非常高。但是，我们的服务器扛不住了。在活动快要结束的最后半小时，购物车和下单结算页面频繁出现页面缓慢，甚至完全打不开、无法下单的情况。很多网友没有抢到想买的书大呼上当。当时，老刘在微博上表示很生气，要求增加 3 倍的服务器，第二天再搞一次促销活动。他还发了一张照片，照片内容是桌子上放着一把刀，并说要请研发部同事们“喝咖啡”。照片中的刀当然是开玩笑的，但是我们心里也知道，这次恐怕不是单单增加服务器就能解决问题了，必须要升级网站的系统架构。2011 年加入京东的架构师小钟接过了这个重任。

整个 Java 版交易系统架构设计过程中，白天各个子模块的负责人分头做接口和业务实体对象的设计，晚上大家就凑一起在会议室里做评审。评审有时非常激烈，常常为了一个属性的名称或者一个接口的参数频起争论。僵持不下时，就由总架构师小钟拍板。评审经常持续到晚上十一二点，我们就叫来外卖，边吃边讨论。感谢麦当劳、肯德基、必胜客风雨无阻的送餐，我们才有十足的精力做项目。

这次架构升级，首先是开发语言由原来的 .Net 换成 Java。开发语言选择 Java 有两个原因。一是国内外非常多的大型网站使用 Java 技术，这是主流，有非常多的经验可以借鉴，也有非常多的成熟开源框架可以使用。二是成本原因，.Net 语言本身虽然不收费，但是 Windows 操作系统是收费的，Visual Studio 开发工具也不便宜。随着公司的发展，服务器越来越多，开发人员也越来越多，这是必须要

优化的。当然就.Net 技术本身来说也有很多优点，如入门快、开发效率高、IDE 完善，而且微软最近也在对.Net 技术做开源和跨平台，.Net 也越来越开放了。硅谷创业教父、《黑客与画家》作者保罗·格雷厄姆说，编程语言是技术和宗教的混合物，如果你想对编程语言进行比较，那就做好准备打一场宗教战。所以 Java 和 .NET 孰优孰劣，各位心中应该有自己的答案。

由于京东的业务发展得非常快，所以为了架构简化、性能分流，需要对原来的集中式系统做切分。把原来集中在一起的系统按业务分拆成了用户中心、商品中心、价格促销中心等。所有系统按 SOA 架构设计，对外提供服务，杜绝跨系统直接读取库。这要感谢负责开发京东 RPC 框架 SAF 组的同事能够提供这样高效、稳定的服务框架。用户中心、商品中心等提供基础信息的服务，构成了基础服务层；其上交易中心，库存中心等，有一些较复杂的业务逻辑的服务，构成了业务服务层；最上面就是购物车、下单结算页、卖场等 WEB 系统。这样拆分后，系统的架构大概呈现下图所示的样子。



京东 2011 年系统升级架构简图

还有存储也做了升级，原来.Net 时代的 SQL Server 要换成 MySQL。数据量大的系统，如订单、商品、用户等，全部要做分库分表。有些大字段从数据库里拿出来，如订单信息快照，商品的详细信息等，放到京东自主研发的分布式存储系统 JFS 里面。

最后是大量使用缓存。在 .Net 时代，京东系统使用的主要是 MemCached 缓存，使用的方法主要是热数据缓存。如商品信息的缓存，在获取某个商品的信息时，先检查缓存中是否存在，如果不存在则从数据库中读取存入到缓存中，在缓存有效期内如果再读取同一件商品的信息就能快速地从缓存中读取了。这样的做法在大多数情况下是有效的，但是如果不幸需要获取一些较冷门的商品信息就会比较慢了。在 2011 年左右，Redis 开始流行起来，京东的系统也开始大量使用 Redis，并且很多关键数据，如商品、用户、价格都是做全量缓存。这样服务的效率就会非常高，如商品提供的服务一次查询一般都在二三毫秒。现在内存也很便宜，堆几台 128G 内存的服务器，再对缓存数据进行精简和压缩，京东数千万条商品信息，数亿条用户信息，也可以全量在内存里存放。

这期间价格系统做了一个非常酷的实现。架构师剑锋出手，开发了一个 Nginx 扩展，从 Nginx 上直接读取存在 Redis 上的价格缓存并对外提供 HTTP 的服务，Nginx 的速度大家都懂得，一下子比原来 Tomcat 提供的服务快了一个数量级。需要调用价格的网页使用 JS 异步调用价格的 HTTP 服务，既实现了解耦，又提高了效率。而后台有个服务监听价格的变化，一旦接收到采销调价或者促销开始结束的消息，后台就会去更新 Redis 缓存中的价格数据。

当然项目实现并不是一帆风顺的。因为时间紧迫（项目的开发时间永远是紧迫的），我们被迫先上了一个半 Java 半 .Net 的版本。即前端 WEB 是 Java 的，后端来不及升级的部分接口以及一些分支业务还采用原先 .Net 的老系统。这么做的好处是我们可以较快地拿出一些成果，坏处是我们要维护 Java 和 .Net 两套系统！而且那部分老的 .Net 接口总是成为瓶颈。后来又过了大半年，我们才把主要业务的 .Net 服务全部替换成新的 Java 版本，而那些分支业务全部替换成 Java 已经是两年后的事了……当交易的最后一个 .Net 系统下线时，一群开发人员聚在一起流着泪说再也不要干两套系统并存这种事了。

新老系统的切换也非常刺激。一个 24 小时都要给用户提供服务的电商网站要升级交易系统，无异于给一列正在飞奔的高铁换轮子，给一架翱翔在两万米高空的飞机换翅膀。为了新老系统的切换，我们在用户将商品加入购物车的入口写了一个非常复杂的切换开关。

一开始先按用户切换，把内部一些用户的账号配置上去，先让少量人员内测。然后再按业务类型切换，先切在线支付的，再切货到付款和自提的；先切自营的订单，再切第三方卖家的订单；先切没有使用虚拟资产的订单，再切使用了优惠券、礼品卡、余额、积分的订单。接下来按地区切换，先把东三省的流量导到新

的系统上去。然后是按数量切换，先切换新系统下 100 单，然后观察一段时间，等到这 100 单最后都顺利地地下发到库房，配送到用户手中，确认没有问题了，再切换新系统下 1000 单……最后是按百分比，首先是 1%，然后逐渐增加到 5%、10%……直到最后切完 100% 的流量，前后竟然花了一个多月。

等到全部切换到新系统上后，我们的一个开发 Leader 祥子夸海口说：“别的不敢吹，就配置开关这个系统，京东绝对业界第一，不服来战。”

在新老系统切换期间，为了验证新系统业务逻辑上的正确性，我们还做了一个比对程序，来验证新系统下的订单和老系统是一模一样的。我们的做法是，在新系统下完单后，马上用一个异步的线程、同样的信息再去调用老系统下一单。然后用新老两个系统生成的订单信息的快照做比对，如果一模一样，则说明新系统的业务逻辑没有问题。比对的方式也很直接，把订单信息的对象序列化 XML，然后用字符串比较。因为 .Net 和 Java 的 XML 序列化后的格式和顺序总有些不一样，也给写比对程序的同事造成了不小的困扰，但这个困扰最后造就了一位正则表达式的大师。当时的情形是，先切换新系统下 100 单，然后一单一单对比，发现那里不一致就让负责这块的同事找原因，改 Bug。改好了，再多放一些订单继续比对。

就这样到了 2012 年 5 月，我们终于把 Java 版的交易系统上线完成了。因为马上就是 2012 年 6 月，为了备战这一年的 618 店庆，我们又做了几件事情。一是添加系统响应时间的监控。所有调用外部服务的、所有对外提供服务的、所有读数据库的，以及一些被认为比较重要的方法统统加上监控，然后每天分析监控中出现的响应慢的原因并逐步优化。这里要感谢开发统一监控 UMP 系统团队做出的努力。二是做线上的压力测试。系统上线前，也在测试环境做过压力测试，但是效果不好。交易系统特别复杂，依赖的服务非常多，压测环境不好搭建，搭建出来也无法反映出线上的真实情况。这种情况下，我们决定在线上环境做压力测试。当然也不可能直接在正在运行的应用上做压力测试，我们的做法是在热备机房做压测。因为热备机房的环境基本和正式环境一样，所以测试的结果基本上能真实反映情况。测试结果出来后，我们都对这一年的 618 店庆的系统稳定性充满了信心。

这一年的 618 店庆当天，系统表现相当稳定。出了一些小小的意外，也都被扑杀在萌芽中了。到了晚上 10 点多，我们闲来无事，都开始赌当天最终会有多少订单了。这之后，前文所述的因大规模促销造成系统宕机的情形再也没有出现过了，老刘也再没有“亮剑”了。

时至今日，除了少数桌面程序，京东的大部分系统已经完成了从.Net 到 Java 的转变。偶尔能在京东网站上看到几个扩展名是.Aspx 的页面，实际上也都是经过 URL Rewrite 的 Java 页面了。（作者/王远）



# 13

## 通天塔——中间件

### 服务框架 SAF

在京东应用系统中使用了大量自主研发的中间件技术，其中在 SOA 领域的关键部件有服务框架 SAF、工作流引擎 PAF、消息引擎 JMQ 3 个。首先介绍第一个服务框架 SAF。

随着京东研发系统的急剧壮大，各个子系统之间的 RPC（Remote Procedure Call）调用越来越多，但是公司内部没有一套统一的解决方案，导致大家使用的技术五花八门，每个系统的开发人员都得考虑如何实现集群化、负载均衡、失败重试等等。这极大地增加了系统之间的复杂度以及开发人员的学习成本。于是 SAF 应运而生。

SAF 是京东全公司统一使用的系统 SOA（Service Oriented Architecture）化的重要基础公共组件，是接入京东开放平台（JOS）的重要前提，它是京东系统服务化、开放化的技术标准。

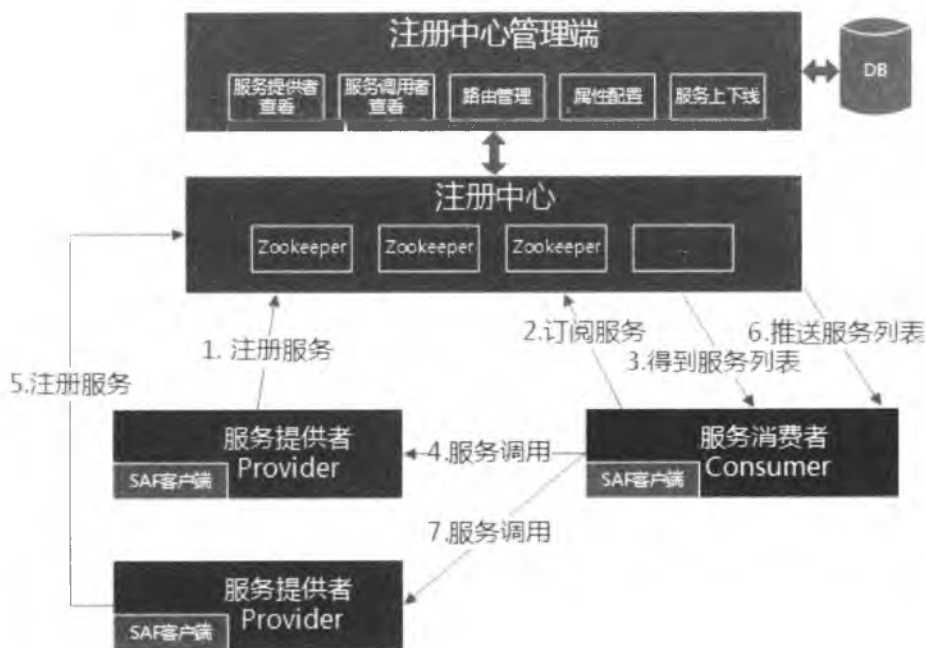
#### SAF 目标

1. SAF 的最终目标是带给全公司一个完整的 SOA 治理功能“生态圈”；
2. 通过 RPC 调用框架收集服务质量数据；

3. 强大的简单易操作的服务管理平台；
4. 通过对数据的分析，可以发现系统的问题所在，也可以计算出最优的服务实例组合，影响 RPC 调用框架；
5. 通过对调用关系的分析，可以勾画出系统依赖图，发现系统耦合情况；
6. 通过对请求量等压力数据的分析，自动伸缩服务集群大小。

## SAF 1.0

SAF1.x 借鉴了业界已有的一些实践，经过了 4 个月就完成上线，提供了高效的点对点 RPC 调用功能以及服务注册订阅功能。后经过几个月的优化，又发布了稳定运行的 SAF1.0.8 版本，到如今已经平稳地经历了多次大促的洗礼。



SAF 1.x 架构简图

## 统一高效的 RPC 调用框架

原来各个系统间的调用有 WebService、RMI、Hessian、REST、Thrift 等等，现在统一为 SAF 协议（兼容 Dubbo 协议）。这样既减少了系统的复杂度，又减少开发人员的沟通和学习成本，提高开发效率。

RPC 调用框架（SAF Client）具有如下特性：

1. 服务调用透明，无代码入侵；
2. 开发人员只需要简单的配置，就可以像调用本地代码一样调用远程服务，专注于自己的业务代码开发，无须关注 RPC 调用的细节（例如，远程通讯、负载均衡、调用的透明化、同步/异步、失败重试等）；
3. 多协议支持；
4. 长连接 SAF、Dubbo、Thrift，短连接 WebService、REST、Hessian 等
5. 实现 Failover，支持失败重试；
6. 当服务调用出现非业务异常，则自动切换 Provider 重试；
7. 多种软负载均衡；
8. 支持随机、轮询、组播等多种负载均衡算法；
9. 稳定高效，默认是使用 SAF 协议基于 NIO 和长连接，TPS 为普通 WebService 的 3~4 倍。

### 简单的服务治理功能

统一了 RPC 调用框架后，大家都使用框架发布 SAF 协议的服务，那么怎么做到服务分布式部署后服务地址的动态发现问题呢？显然仅仅 RPC 框架是不能完美解决的。

我们引入了一个“注册中心”的概念。注册中心保存了服务的提供者列表及配置信息。服务提供者启动的时候去向注册中心注册服务地址，服务调用者启动的时候会去注册中心订阅服务列表，服务调用者拿到列表后才向服务提供者发起 RPC 调用请求。

第一代 SAF 的注册中心是基于 Zookeeper 开发的，结合自主研发的管理平台，主要实现了如下服务治理功能：

1. 服务提供者的归属信息；
2. 服务的注册和订阅功能、变化动态推送；
3. 服务分组、上下线功能；
4. 简单的服务路由功能。



SAF 1.x 管理界面

目前线上使用规模如下：

1. 发布接口 1620+;
2. 接入服务器按独立 IP 计算 2700+;
3. 接入 JVM 实例 21000+个;
4. 接入服务提供节点 22000+;
5. 接入服务调用节点 63000+;
6. 其中营销研发部约占 42%;
7. 其中 500+接口接入 JOS (京东开发服务)。

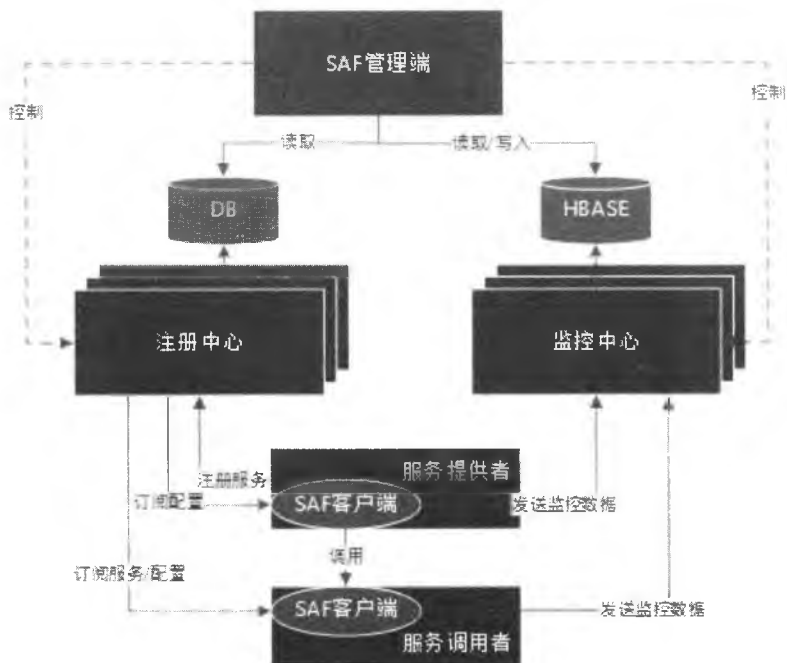
## SAF 2.0

时间又过了两年，第一代 SAF 稳定地运行着，不过随着接入量的攀升，以及京东系统 SOA 化的不断演进，慢慢地我们还是发现了一些问题，具体如下：

1. 目前的 SAF Client 逻辑任务较重，很多逻辑可以在注册中心完成，这样保证客户端的轻量化，使其性能更佳。
2. 使用 Zookeeper 注册中心的局限性。包括树状结构导致无法从多维度对数据进行遍历检索，无法实现业务逻辑的预处理，Leader/Follower 模式导致无法做到水平动态扩容。
3. 服务质量管理功能的缺失。目前无法得到调用量、并发、流量等统计数

4. 服务治理功能较弱,需要更加强大的服务治理功能。例如,调用端归属,动态分组,限流,黑白名单,请求优先级等。

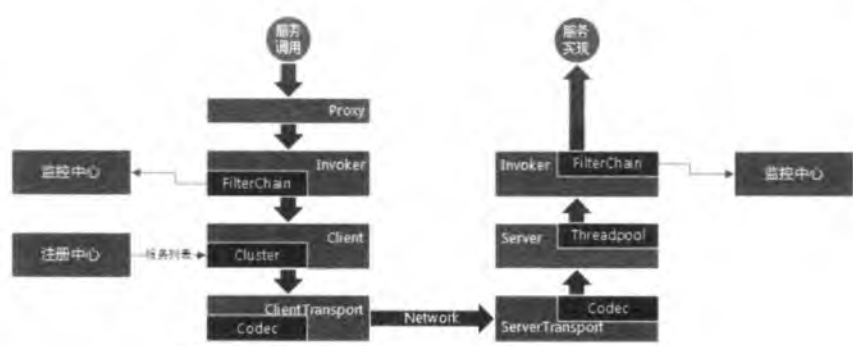
为了解决这些问题，新一代的 SAF 已经开始启动，SAF 2.x 的各个模块组件完全自主研发，保证代码灵活可控、高度定制。



### SAF 2.x 架构简图

### RPC 模块特点如下:

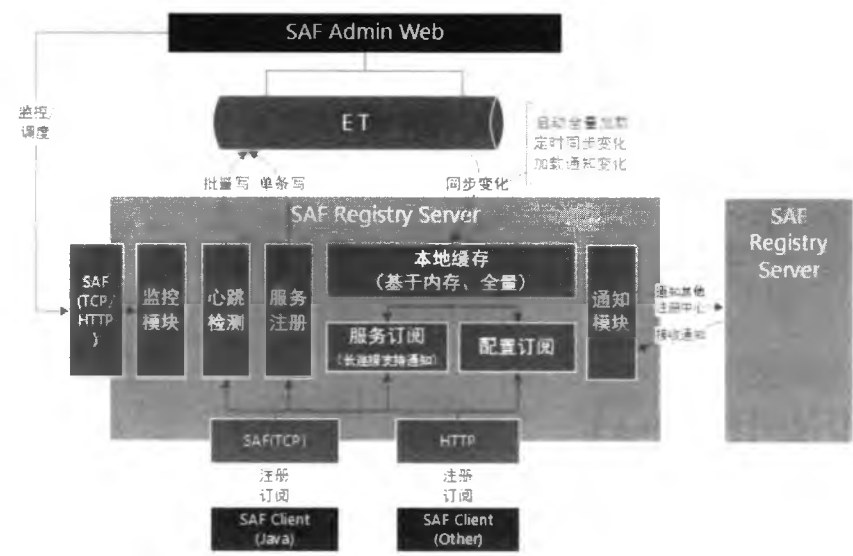
1. 轻量、稳定、高效；
2. 将计算逻辑放在注册中心，客户端尽量只关注结果，不再运算数据；
3. 高可扩展性；
4. 兼容新旧版本互调，提供丰富的接口切入点供业务扩展使用；
5. 服务质量数据实时收集；
6. 按分钟级、实例级、方法级去统计调用数据，根据配置规则实时发送给监控中心。



SAF 2.x RPC 框架简图

注册中心特点如下：

- 1. 无 Master、水平扩展、稳定高效；
- 2. 灵活的调用方式，数据推拉结合；
- 3. 包含业务逻辑，减少客户端负担。

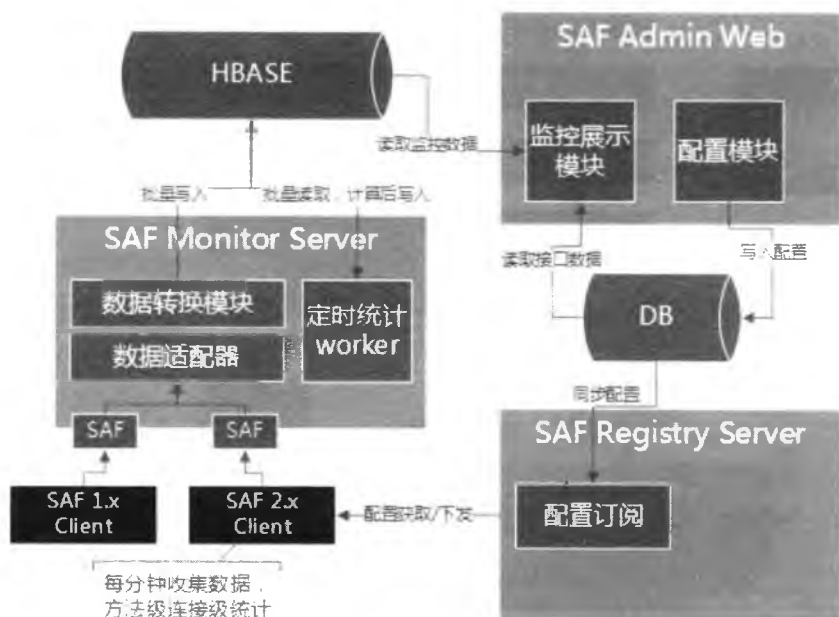


SAF 2.x 注册中心架构图

服务质量管理特点如下：

- 1. 细粒度的数据收集，兼容新旧版本的数据格式；
- 2. 灵活自动的数据采取策略；
- 3. 多维度的数据展示和报警功能；

4. 通过 QoS 计算服务配置，下发到客户端。



SAF 2.x 服务治理管理架构图

## 工作流引擎 PAF

### PAF 介绍

PAF (Process Architecture Foundation) 工作流平台。对于大中型公司来说，工作流是必不可少的。从业务的角度来说，流程代表了自动化和标准化；对业务系统开发人员来说，工作流可以减少流程开发的工作量，缩短流程变更维护的时间，降低系统中 Worker 的使用率，尽量避免系统中到处使用业务规则判断。

PAF 应用场景如下：

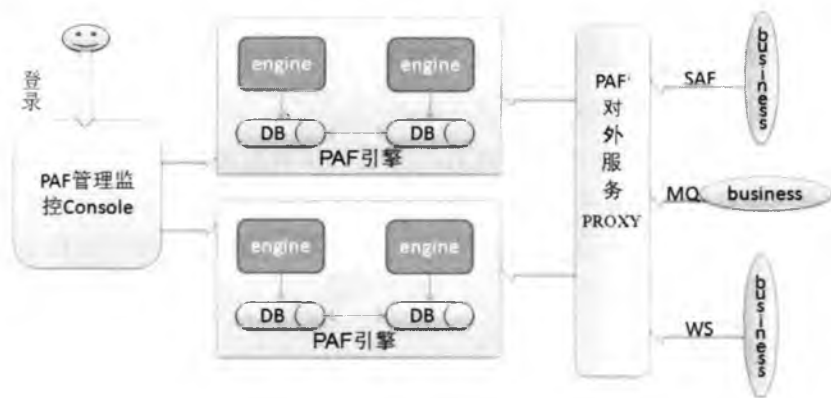
1. 项目中有一些或许多人工审批流程，比如礼品卡/优惠券审批，等等；
2. 项目中存在比较频繁变化且复杂的业务逻辑及流程；
3. 项目中需要按一定的顺序和许多的外部依赖系统进行同步或异步的交互；
4. 项目中存在大量复杂常变的业务校验规则。

PAF 可以给业务系统带来如下好处：

1. 减少流程开发的工作量；
2. 缩短流程变更维护的时间；
3. 降低系统中 Worker 的使用率；
4. 尽量避免系统中到处使用业务规则判断。

## PAF 创建

PAF 创建时，遵循了“拿来主义”，基于开源工作流系统 Activiti，搭建了专属京东内部的工作流平台。Activiti 官网上的“*As Activiti is ‘just a jar’*”是吸引我们的一个重要原因，虽说 Activiti 支持分布式部署，但由于其核心引擎只是一个 jar，这就给我们搭建平台提供了相当大的方便。经 PAF 组老大周架构的简单设计，Activiti 的水平扩展能力得到相当大的提高。



部署架构

最外面封装一层 PROXY，可以接收多种调用方式。PROXY 内部设置了路由规则，可以把不同的业务系统调用分配给相对应的 PAF 引擎。大家会发现，PAF 引擎功能还是很强大的，可以无限制地水平扩展。PAF 引擎一般对应一个业务系统，但如果业务系统流程量不是很大，可以多个业务系统共用一个 PAF 引擎，如

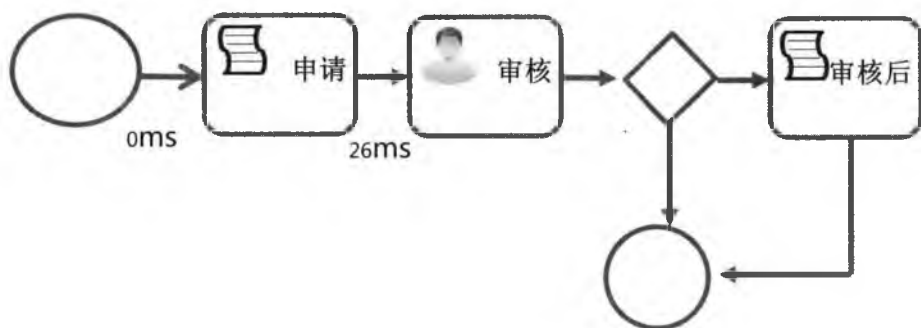


果业务系统的流程量很大，可以给该业务分配多个 PAF 引擎。

## PAF 发展

PAF 上线后，第一个接入的是采购单审批及回传业务，这是一个纯自动的流程，不需要人工干预，按规则调用业务系统的几个接口即可。所以，PROXY 仅提供了一个接口，一个方法，用户只需启动流程，超级简单。

业务使用过程中，会时不时查询一个实例的当前运行节点及执行轨迹。这时 Activiti 提供的东西就不够了，它只提供了当前节点的显示。所以经过小李同学的设计后，我们可以作到如下展现：



流程运行示意图

从图中可以直观地看到流程运行到哪了，走了哪些节点，直观方便。

工作流系统人工方面最直接的应用就是 OA 系统。公司内部的 ERP 打算接入 PAF 时，提了 N 多的非常个性的需求，比如，IE 版流程定义设计器、模拟计算下一节点、主动拉回，等等，这些特性 Activiti 都是没有的，所以只能自己实现了。其实这也是引进国外的工作流系统的一个弊端，他们支持不了国内特有的这种需求，是不符合国情的，也可以说“水土不服”。

## PAF 现状

PAF 上线后，先后接入了采购单审批及回传业务，金融的多个流程，ERP 系统，等等。现在线上环境的流程定义近 170 个，流程节点数量平均 15 个，每天的流程实例近 100000 个。

## PAF 存在的问题及持续改进

通过对自动流程的性能测试，我们也发现了 Activiti 自身的一些问题。例如，

测试流程中,我们设置了 8 个 ServiceTask 节点用来调用外部的 WebService 服务,同时设定外部服务的响应时间为一秒,性能测试结果如下:

请求总数	请求失败数	请求TPS	请求平均响应时间	流程TPS	流程完成平均时间(毫秒)
141097	0	38.248	0.077	38.53	12559.96
157542	0	42.691	0.641	42.64	14582.48
175661	0	47.796	0.656	43.27	14599.648

性能测试结果

测试结果显示,单个 DB 所能支持的流程实例的最大 TPS 才 40 个左右,各服务器 CPU 及内存都很空闲,只是 DB 服务器数据库连接数达到了最大值。通过分析发现,Activiti 在执行 ServiceTask 节点时,调用外部服务的时间也会 hold 住 DB 的 Connection,导致 DB 连接一直不释放。

发现了问题,并且找到了原因,剩下的就相对好办了。是持续对 Activiti 进行改造还是自研一套新的 workflow 引擎,这就是我们下一版本主要思考并解决的问题了。

# 京东消息中间件 JQ

消息中间件是公司信息系统 SOA 化进程中的一个重要中间件产品,可用于系统解耦、非实时系统通信、异步流程处理等。消息中间件主要经历了 3 个阶段。

开始时由于使用的范围并不广,消息量也不大,开发了基于数据库存储的 JQ,随着使用范围越来越广,JQ 的性能问题和数据库存储压力日益凸显。

为了解决 JQ 暴露的问题,需要抛弃数据库存储,为了快速上线并能够保证稳定性,公司选择了使用范围非常广泛的 ActiveMQ 进行定制开发,实现了一个

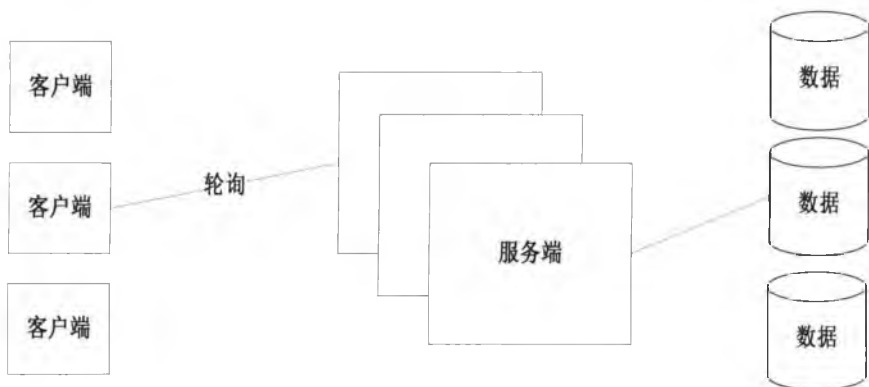
分布式的客户端和全新的管理平台，而 ActiveMQ 一些功能的改进则将在运维过程中完成。

公司业务量飞快增长，日均消息量突破 10 亿。现有消息平台 AMQ 已无法满足业务增长的需要。VirtualTopic 会根据订阅者数量复制多份消息，导致发送性能低；Slave 不支持消费，当一组内的 Master 宕机，又不满足切换条件时，积压的消息无法及时被消费；消费确认需要删除索引，导致消费和生产竞争相同的锁，影响发送性能；消息积压量比较大时会严重影响发送性能；AMQ 的处理能力满足不了一些高吞吐量的系统需求，比如访问日志收集等。为了解决这些问题，更好地满足公司的业务需要，我们全新开发了新一版的消息平台——JMQ。

下面详细介绍一下各个版本消息中间件的技术细节：

## JQ 时代

JQ 是一个基于数据库的消息系统，服务端负责管理消息的路由，客户端通过定时轮询从数据库加载消息，消费后的消息会定期从数据库中删除。



JQ

## 缺点

1. 客户端采用轮询的方式查询是否有新的消息，影响消息到达消费者的及时性，如果查询频繁又会给数据库带来比较大的压力；
2. 当消息发生积压时，数据库存储压力比较大，也会影响其查询性能；
3. 受限于数据库的性能，不能满足分布式的、大规模的消息生产和消费。

## AMQ 时代

基于 ActiveMQ 进行二次开发，对客户端进行了大量的改造，使其支持分布式部署，能自动感知服务端集群变化。开发了功能完善的管理程序，提供了消费异常消息自动重试功能，增加了报警、发送和消费审计日志查询等功能。

同时，对 ActiveMQ 的 Broker 也进行了大量的改造，重写了复制协议，优化了 VirtualTopic 写入性能等。

该产品具有如下特点：

### 1. 技术标准

- 遵循JMS1.1规范；
- Zookeeper作为分布式协调器。

### 2. 高可用

- 采用Master/Slave模式部署，支持Failover；
- 采用同步发送、消息持久化，防止数据丢失；
- 分布式Java客户端，支持多组MQ服务器负载均衡和Failover；
- 消息异步归档到云存储，便于故障跟踪；
- 统一的异常消息重试服务，便于容错。

### 3. 扩展性

- 按消息类型垂直划分；
- 分布式Java客户端，支持水平扩展、动态扩容、负载均衡。

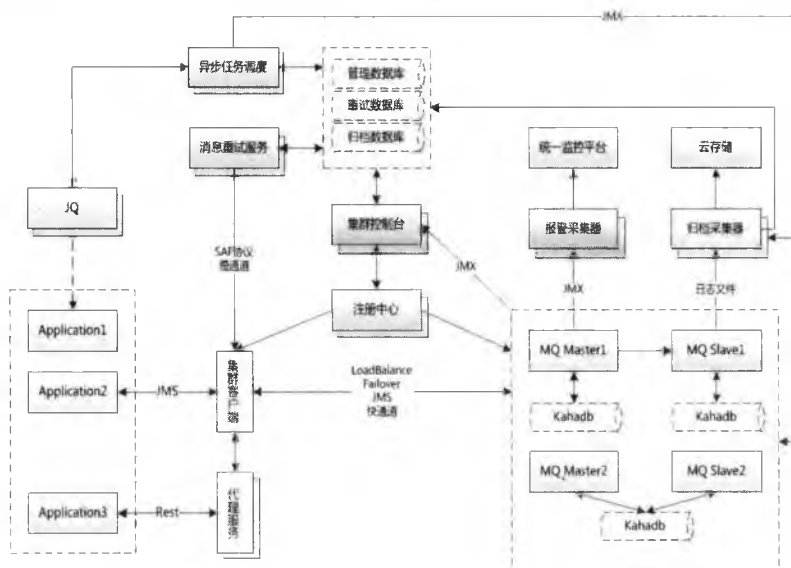
### 4. 管理监控

- 集群的管理控制平台，包含消息类型管理、发布订阅管理、业务系统管理、服务器管理、归档查询、异常消息管理、运维计划，等等；
- 支持积压报警、存活报警、内存占有报警、磁盘占有报警、连接断开报警、主从状态报警、报警查询；
- 消息定时统计邮件报告。

### 5. 易用性

- 集成了消费者异常消息重试服务，业务系统主要关注于业务逻辑的处理；
- 兼容JQ的客户端接口。

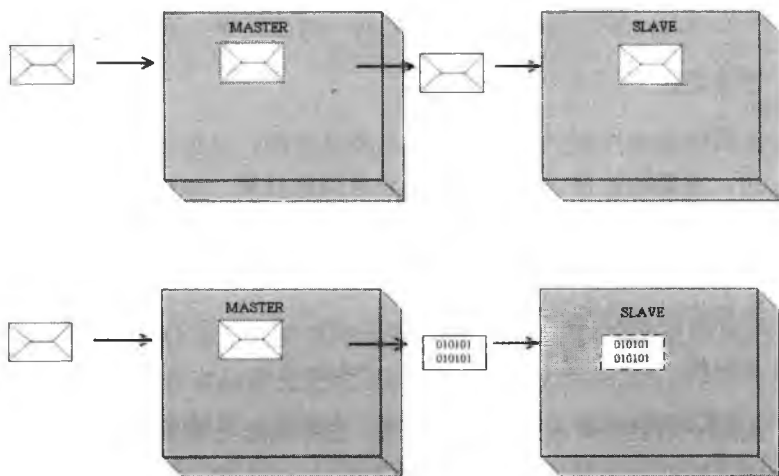
整体架构如下图所示：



整体架构图

## 复制优化

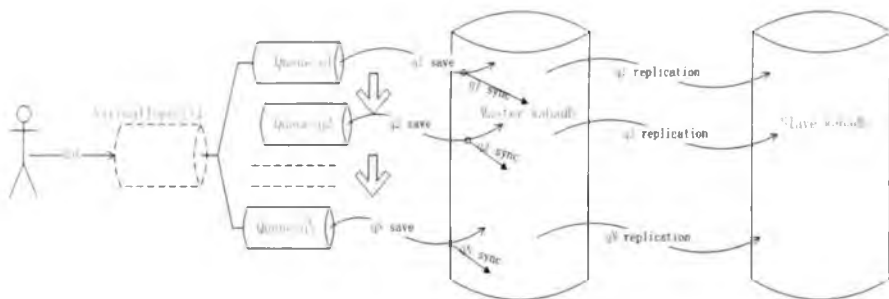
对 ActiveMQ 原来基于消息转发的方式进行 Master-Slave 同步复制的做法进行了改造，改为直接基于消息日志进行复制。



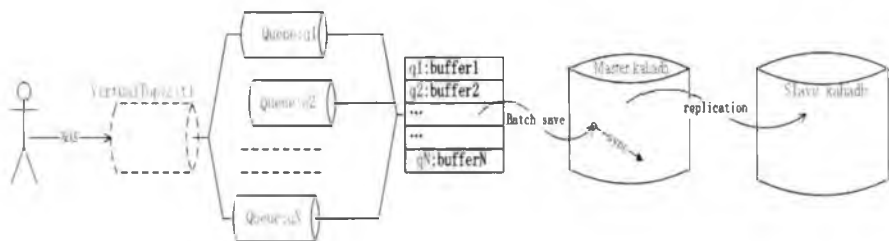
改造前后对比图

## VirtualTopic 优化

VirtualTopic 接收到消息后，会内部转发消息到订阅者的消费队列上。有多个订阅者时，一次发送会多次写入与刷盘，通过合并做到一次发送、一次写入、主从复制时间也只需要传输一次，提高了发送性能。



优化前示意图



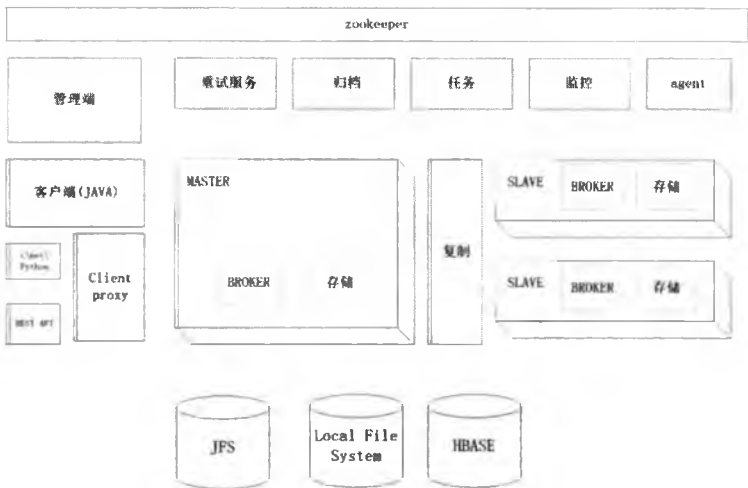
优化后示意图

### 缺点

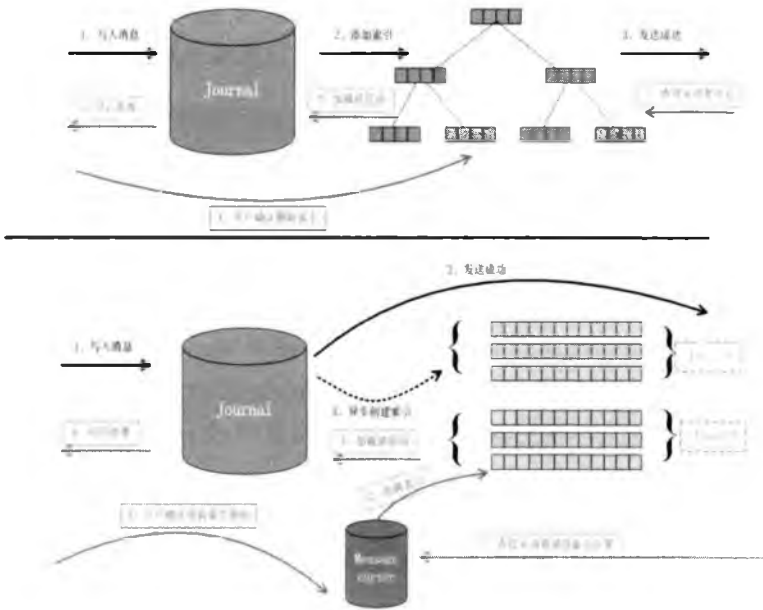
1. 存储引擎 KahaDB 索引使用了 B-Tree 结构，消息发送和消费确认时都需要对索引进行改动，发送和消费相互影响；
2. 由于语义的限制，Topic 有多个订阅者，如果订阅者内部需要支持多线程负载均衡和快速的 Failover，则需要使用 VirtualTopic。VirtualTopic 为每一个订阅者创建了一个独立的 Queue，消息会复制到订阅者对应的 Queue 中，随着订阅者增加，消息需要复制的副本数也越来越多，严重影响发送性能；
3. Slave 只能进行数据的备份，不能承担消费功能；
4. 客户端逻辑复杂，新增功能一般都涉及客户端版本的更新，不利于推广。

JMQ 时代

JMQ 吸取了 AMQ 的一些经验教训，对存储模型进行了重新设计，整个应用完全自主研发。除保留了上一代 MQ 的特性外，还提供了一些新的特性，同时性能比上一代更好。



整体架构图



存储变化图

原来存储的特点如下：

1. 同步建索引，B-Tree 结构；
2. 索引共用有锁，多队列共用索引文件。
3. 消费影响生产，消费确认时需要修改索引、删除已消费节点，需要占用索引文件锁，影响发送性能。

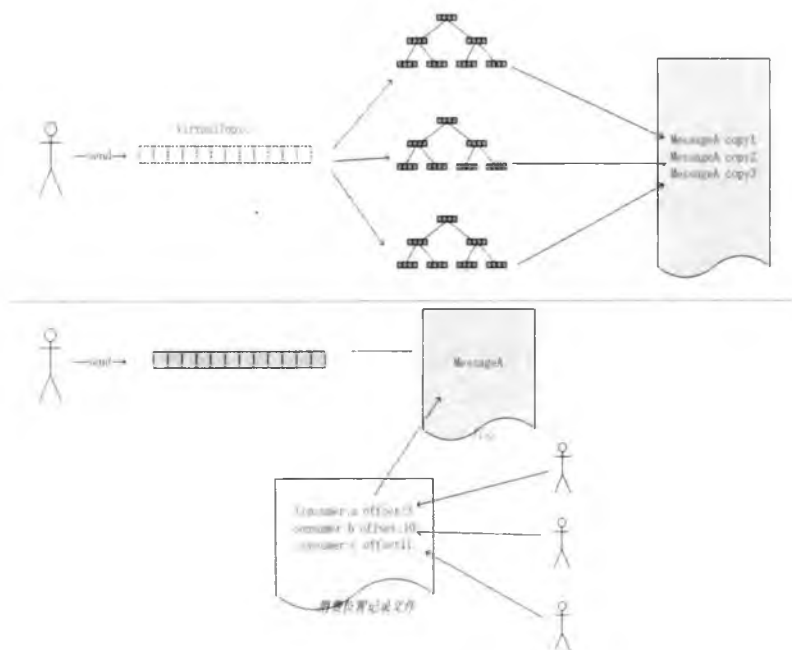
现在存储的特点如下：

回：

1. 异步建索引，LIST 结构，顺序追加，发送阶段写入日志成功后直接返
2. 消费不影响生产，消费确认时只修改索引游标位置，不需要改动原索引结构，对发送性能无影响；
3. 消费位置可以重置，实现回放。

### Topic 变化

当 Topic 有多个订阅者时，通过单独记录每一个订阅者的消费位置，实现了多订阅者共用同一份索引和消息体，同时也支持单个订阅者多线程消费。



AMQ 与 JMQ 关于 Topic 的对比图

(作者/李鑫、章耿、丁俊、桂创华、王松林、梁秋实)



# 14

## 变形金刚——SOA治理

### 服务化开端

随着京东的飞速发展，原有的技术架构已经难以支撑业务的爆炸式增长了。因此，我们踏上了 SOA 的探索之旅。京东的 SOA 实践，是一个不断发现问题，解决问题的过程；也是一个由合而分，又由分而合的过程。而中间件与开源技术的引入，对其产生了巨大的推动作用。

京东技术发展早期，业务系统大都直连数据库。不但简单可靠，而且开发效率高。进入 2009 年，随着业务的快速增长，大量的新系统喷涌而出。系统之间、系统与数据库之间，相互影响的现象越来越严重。以订单为例，商城业务都以订单为驱动，因此每个系统都要包含一份订单的逻辑代码。一旦某个系统的订单逻辑错误，就会导致订单产生“奇妙”的行为，如订单已经取消了，商品还会被送到顾客家中。而且新功能上线时，系统出错的概率也越来越高。经常出现修改了交易代码，库存出问题了，或者修改了商品代码，交易出问题了等现象。记得那时要经常修改交易数据，开发人员和 DBA 经常彻夜工作，苦不堪言。



2009年改造前的架构示意图

为了解决这些问题，Donny 提出了建立 5 大业务中间件（不同于消息中间件等通用中间件）的构想，并带领大家对这一构想进行了实践。业务中间件将业务系统中的公用逻辑抽象出来封装在一起，对外提供统一接口，供业务系统调用。例如其中的订单中间件——它对订单的状态变化规则进行了抽象和封装。订单的状态变化是有规则的，只有满足了指定的前置状态，才能进入后续的状态。比如，只有正常的订单才能进入库房生产，而取消的订单不能进入生产配送环节；京东的订单状态有几十种之多，规则异常复杂。而这些规则，业务系统不可能也无须知道，需要时只要调用订单中间件即可。



2009年改造后的架构示意图

这样做有几个好处：首先是模块复用，如需通用功能，只要调用业务中间件即可，开发难度大大降低；其次是系统解耦，通用功能发生变化，业务系统均无

须改动，反之，业务系统发生变化，中间件也无须关心，开发效率明显提升；再次是应用与存储解耦，业务系统是禁止操作其他系统的数据表的，必须通过业务中间件完成，从而在根本上避免了上述订单状态“乱跳”的问题。

然而，上述方案也引发了一些新的问题。因为是远程调用，性能有损耗。而且任何一个中间件系统不可用，将产生“蝴蝶效应”，导致大量的系统都不可用。在上线之初，就因为网络环境的问题，以及资源管理不善，导致了服务短暂的不可用，造成了一些不良后果。这件事使得当时一些人对服务化充满了疑虑，延缓了京东的服务化进程。

## 服务拆分

2010年，京东进军图书市场，几次图书促销，搞得异常火爆。然而，系统的压力已经渐渐超出其最大容量了。为了支撑订单的爆炸式增长，主交易系统的架构升级已经刻不容缓。于是，一个声势浩大的“改造运动”开始了。

首先要做的就是分离主交易系统和主数据库。将用户下的订单保存到主库的过程是一个长事务，需要200ms，而且主库是写库，无法水平扩展，于是我开发了一个接单服务。顾名思义，它的职责就是接受并保存订单。但并非保存在主库，而是在自己的接单库中。接单库是对等的，可以水平扩展，理论上容量没有上限。但还有一个问题，就是订单号。订单号必须同步生成，不能重复（以前是订单主表的自增主键，接单库变成多个之后，就无法使用了），其他所有系统都是用订单号和交易系统进行交互。为此，我又开发了一个订单号服务。它利用模同余的等价关系，保证订单号不会重复，大致地单调递增，并且足够可靠，支持水平扩展。

从主交易系统陆续分离出来的还有支付系统、促销系统、库存系统、商品系统、券卡系统、用户系统等（5大业务中间件只是封装了部分公用逻辑，此次才是真正的业务分离）。与订单类似，这些系统也将自己的存储从主库中逐渐迁移出来。



服务拆分后的架构示意图

改造后的系统，性能有了大幅提升。还以保存订单为例，需要的时间从之前的 200ms 降到了 10ms。更重要的是，系统获得了水平扩展的能力，只要简单地加几台机器就可以使得系统容量成倍增长。而且还具备了容灾的能力。哪怕大部分服务或者数据库宕机了，只要还有一组（一台应用，一台数据库）可用，系统就可以正常运行（其实在当时一组机器就可以支撑全部的访问量，只是出于容灾考虑，才配备了 4 组机器）。这个容灾还支持跨机房，即使整个机房宕机了，也可以立刻切换到另一个机房，保证系统的正常运行。

这次架构升级，是京东继 2008 年封闭开发之后的又一次浴火重生。其惊险程度好比“在一辆高速行驶的赛车上更换轮胎”。因此在实施的过程中，必须谨慎地处理每一个问题。

首要解决的问题就是系统如何平滑迁移。京东是互联网公司，一切以用户体验为优先，不能因为升级系统而停机，即“在更换轮胎的同时，赛车仍能高速行驶”。说起来容易，做起来难。系统进行了大规模重构，如何保证重构前后的系统行为是不变的？新系统繁多，不可能同时发布上线，又如何保证新老系统之间是兼容的，可以并行运行？而且这必然是一个长期的过程。再有，重构前后，数据结构发生了变化，新旧数据之间如何并存？

为了解决应用层的平滑迁移问题，我提出了一个两阶段切流量方案。第一阶段：**按照用户切流量**。即设置白名单，只有指定的用户，才能进入新系统，其他用户只能走老系统。主要用于上线初期的内测阶段。第二阶段：**按百分比切流量**（为了保证数据的一致性，可根据不同的业务场景，采用不同的百分比方案。例如，订单号取模，或者对用户账号取哈希值再取模）。先让万分之一甚至更少的用户进入新系统，观察有无问题。如果有问题，则立刻切回老系统，待解决问题后，再切流量到新系统。如果没问题，则逐渐放开流量，1/1000、1/100、1/10、1/2，直到全部打开。这主要用于上线后期的公测阶段。

这里还有两个问题要解决：一旦出现问题，如何迅速察觉和定位？又如何快速切换到老系统？要在如此众多的系统中，动辄几百台机器里，快速定位和切换谈何容易。在当时，京东并没有成熟的监控系统。于是我们自己写了一套监控系统，用于快速定位，一套配置管理系统用于快速切换。监控系统比较简单，在 Log4j 中加一个 Appender，将系统的错误日志和性能参数打到缓存或者 MongoDB 中。在管理界面上，就可以近乎实时地观测到系统的状态和异常信息。而配置管理系统，就好像水流的开关，可以在 30 秒之内，随时打开或者关闭几百台机器的新系统入口，或者调节进入新系统的流量大小。

在解决数据层平滑迁移的问题时，Andy 提出了四阶段切换方案。第一阶段，同步写老数据库，异步写新数据库；第二阶段，同步写新老数据库；第三阶段，异步写旧数据库，同步写新数据库；第四阶段，同步写新数据库。在这个过程中，还要根据具体的业务场景，制定系统回滚方案，确保生产环境不会因为系统升级而受影响。

无论应用层、还是数据层，迁移过程中都需要验证新旧系统之间逻辑和数据是否一致。除了测试之外，还需要数据校对（逻辑的处理结果反应在数据上）。我们将新老系统的数据进行了逻辑校对。这个过程持续了很长时间，直至上亿次的校对都正确之后，才逐渐放开流量。

后来这两条经验传播到了其他团队，如今已经成为京东内部做系统升级时的常用方法。这只是架构升级过程中的两段插曲，我们在实践中，总结出了许多经验法则来保证系统的平滑升级。（如果有机会，希望能将这些经验法则归纳出来，以飨各位读者。）

而这都需要非凡的精力、毅力才能做到。其中的苦涩艰辛更非语言所能描述。记得封闭开发的那段时间，Andy 的女儿刚好出生，他每天要跑 3 个地方：公司、家里、医院。有一次实在太累了，他竟然在地铁里睡着了，但我们却没听他抱怨过一句。封闭开发的办公室里没有暖气，他而当时正值寒冬，然而大家的工作热情，却使得办公室里异常火热。不过，当时也有一件令人期待的事情，就是中午大家一起去旁边的小店里吃一条热气腾腾的烤鱼。从服务员端上来到结束战斗，基本不超过 5 分钟。前几次服务员都很吃惊，后来就见怪不怪了。

此次架构升级，对于京东信息系统而言，是一个划时代的转变。正因如此，我们才能扛过一个又一个的 618 和双 11 庆典，系统仍能岿然不动。京东在 SOA 这条路上，已经扬帆起航！

## 服务治理

2011—2012 年，京东老系统不断拆分，新服务如雨后春笋般冒出来。系统拆分之后，有诸多好处。每个系统和服务，都可以为其他系统提供可重用的模块，便于开发新的业务；可单独部署，并能随时扩容；系统之间是业务松耦合的，一个系统的修改并不会影响另一个系统；项目的开发周期缩短，等等。然而，新的问题又出现了。系统之间的交互和依赖关系异常复杂，以至于后来没人能说得清楚，而且每天都在变化。系统的依赖越来越多，远程调用越来越频繁，网络开销也就越大，可用性和性能变差。系统的“蝴蝶效应”越来越明显，尤其是底层的服务和系统。它们稍一抖动，上层系统就会产生震荡，如果其中之一宕机，会导致大量的系统不可用，甚至整个系统崩溃。因此，如何管理和监控这样一个高度分散的服务化环境，已经成为迫在眉睫的任务。

后来从支付宝来了一个牛人——晓钟。他带来了支付宝的分布式服务框架 Dubbo（后来在此基础上衍生了京东的 SAF 框架）。该框架解决了服务治理的问题。

服务提供者向注册中心注册自己的服务地址；服务消费者向注册中心查询自己感兴趣的服务提供者的地址，然后根据获取的地址，向服务提供者发起请求；服务提供者和消费者，都向监控系统提供自己的运行情况。这 4 种角色，没有任何一个是单点（服务提供者和消费者是一个集群），任何一个宕机，都不会对整个系统造成影响。而且服务之间的依赖关系由注册中心实时记录，并如实地反映到管理系统中。运维人员可以对其进行监控和管理。例如某个系统上线新功能，会影响哪些系统呢？可以由系统向依赖于该服务的系统负责人发送邮件或者短信进行提醒。监控系统可以根据服务提供者和消费者的运行情况以及依赖关系，对系统进行控制和负载均衡。例如服务提供者的一个集群响应时间变长，则监控系统会自动将更多的流量分流到另一个响应时间更快的集群。新项目发布时，则可以通过监控系统，对某个集群进行上下线控制，从而使得系统平滑迁移。如果某个服务提供者已经达到了其系统容量上限，则监控系统可以对其进行限流保护，防止其崩溃。



引入服务框架后的架构示意图

至此，京东系统由合而分，又由分而合，已经基本完成了服务化进程。然而还有几个问题需要解决。其一，数据复制延迟的问题；其二，如何将松散服务，有效地组合成业务流程，以便更有效地重用；其三，如何解决服务的级联反应问题。

## 事件驱动

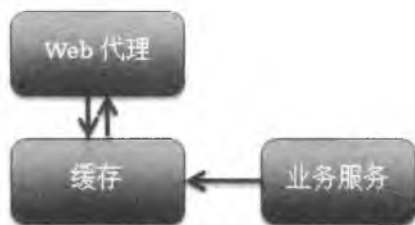
数据复制延迟问题，一直困扰着京东。这里举两个例子。

一个是价格问题。京东经常降价促销，引得大量用户抢购。有些用户将商品放进购物车，在提交订单的时候，却发现价格已经恢复了原价，原因在于缓存。价格服务每天的访问量在几十亿以上。为了支撑如此大的访问量，系统设置了多级缓存，包括页面缓存、CDN 缓存、服务器端缓存，等等。而缓存是有过期时间的，设置得过短会增大系统压力，设置得过长则会造成上述的结果。

为了提升用户体验，我们决定将延迟时间从分钟降至秒级，于是“实时价格”项目启动。通过事件驱动机制，降低数据延迟时间。改造前后的架构示意图分别如下：



改造前的实时价格架构示意图



改造后的实时价格架构示意图

改造之后，Web 代理层直接从缓存中获取数据，比原来从业务服务层获取数据要快一个数量级，大幅提升了系统性能。一旦触发了一个业务事件，应用系统就会更新缓存中的数据，成功将复制的延迟时间缩短到了秒级。后来去除了 CDN 缓存，价格系统仍然坚如磐石，成为名副其实的“实时价格”。

另一个是订单生产问题。用户下单之后，订单要进入一系列的生成流程。当时从下单到订单进入库房生产经常需要 20 分钟，并且一旦数据库复制延迟，就会造成大量的订单积压而无法生产。为了提升仓储生产效率、降低运营成本，研发部启动了另一个更大的项目“211 订单履约”。该项目将订单生产主流程进行了彻底的改造，用户一旦下单成功就触发一个业务事件。该事件会将订单从下单、拆分、转移、发票等环节一直推送到进入库房生产。时间从之前的 20 分钟，缩短至 10 秒左右，极大地提升了订单生产效率以及研发部的技术地位。该项目以其牵涉之广、改造之复杂、影响之大，被列为公司 2012 年 10 大项目之首。





引入事件驱动后的生产系统架构示意图

事件驱动使系统具备了快速响应的能力。如果把 SOA 比作手和脚的话，那事件驱动就像人的眼睛和耳朵。当眼睛发现一只狮子正朝你奔来时，一个消息被发送到大脑，然后大脑向你的手脚发出指令：赶快跑。其次，事件驱动将引用解耦。在交互过程中，并不要求各个服务都同时在线。如果某个服务出现短暂的不可用，并不影响其他系统的正常工作。例如 OFW 系统会将数据暂时存储起来，等该服务可用之后再进行处理。因此，它还具有抵抗洪峰的作用。另外，事件驱动可以实现流程编排，将在下一节详细讲述。

后来引入的消息中间件 MQ，则将事件驱动机制以更简单的方式带给了其他系统。例如财务对账系统，只要应收款项或实收款项发生变化，就会触发对账流程，处理时间由分钟级缩短到了秒级。

## 流程编排

2013 年年初，公司为了提升用户体验，启动了另一个影响更为深远的项目“SOP 合并下单”。即用户把京东自营的商品和第三方商家的商品放在一起下单，

同时,告诉用户订单由几个包裹组成以及配送情况。这相当于要在用户下单的一瞬间,将订单的主要生产流程执行完毕后,展示给用户观看。这牵扯到几乎所有主要系统,如交易系统、财务系统、拆分系统、转移系统等。因为系统众多,还要考虑服务的级联反应问题。如果一个服务宕机,将导致用户无法查看自己的订单情况。

订单结算页

订单结算页

## 提交订单

背后的故事:券、卡、余额、库存系统、Promise 系统、商品系统、促销系统……

订单成功页

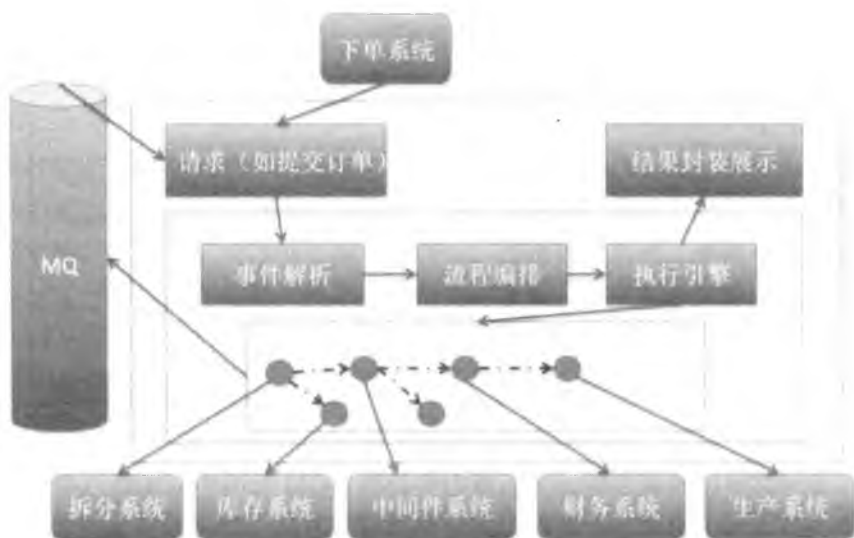
订单成功页

## 订单成功提交

背后的故事：拆分系统、订单系统、库存系统、OFW 系统、财务系统……

为此，我将事件流程引擎引入到交易系统中，取名为“交易引擎”。它将各大系统组合在一起，带动了生产流、资金流、信息流的运转。

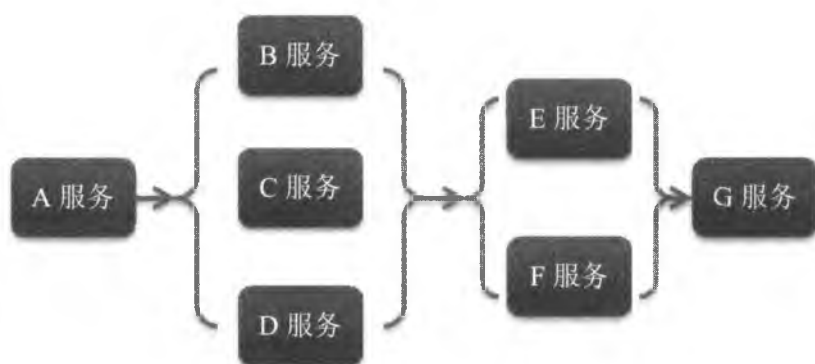
首先，它接收到下单事件后，对事件进行解析；然后，对流程进行编排，根据业务规则判断哪些调用有因果关系必须串行，哪些调用可以并行执行；接着，按照分析结果开始调用各个服务；整个流程执行完毕后，将结果封装展示给前端。架构示意图如下：



交易引擎架构示意图

由上可见，一个事件可以引发几个服务被调用。反过来一个服务也可能产生多个事件。事件根据因果关系被不断地传播和执行，最终形成业务流程。以前松散而独立的服务通过事件和流程组织起来以后，就构建出更高层次的解决方案，就像搭积木一样。而这正是 SOA 的目标。

另一方面，它较好地缓解了服务的级联反应问题。首先它将某些服务的调用改为并行执行。



服务并行调用原理图

整个流程的响应时间将是 A 服务时间+BCD 服务中最长的时间+EF 服务响应最长的时间+G 服务时间。而原来则是所有服务时间的总和。其次，抛弃了以前一请求一线程的模型，而是采用类似于协程的方式，将线程与请求解耦。从而避免了因为 RPC 调用密集而导致的网络连接数飙升，有效地遏制了服务故障在整个系统中的扩散，即“蝴蝶效应”。

以上只是波澜壮阔的改造史诗中的几段插曲，背后耐人寻味的故事更是数不胜数。然而对于京东来说，这仅仅是序章，O2O、大数据、云计算等正方兴未艾，京东正在迈向新世代。（作者/赵海峰）



# 创新

京东有哪些应用创新？京东是怎么激发创新的？

本篇将从移动创新、大数据应用创新、创新孵化管理机制3个方面展示京东研发在应用创新领域的探索。

看京东如何实现在手机上试戴眼镜、试穿衣服，感受大数据是如何实实在在地驱动并优化着京东的核心供应链，怎样支撑采购品类决策、定价决策、促销决策、库存决策、物流路径决策等。

- 云平台
- 移动创新
- 大数据洞察未来
- 智慧的采销
- 创新孵化



# 15

## 壮志凌云——云平台

### 京东云背景

云计算，是 IT 产业继个人计算机变革、互联网变革之后的第三次变革。

电子商务以 IT 信息系统为核心，这使得电子商务更需要云计算的支持。对内，云计算从开发、运行、运维 3 个层面降低系统成本，提升系统稳定性，提高电商运营效率。对外，云计算的 SaaS 服务可以向上游卖家和下游买家提供丰富的电商应用。电子商务的飞速发展向云计算提出了众多的业务需求和技术难题，这些也引领着云计算向前发展。

京东云平台成立于 2012 年 4 月，是京东研发系统的一级部门。2011 年，京东确立以技术为驱动的总体战略——未来十年京东将发展电商平台、物流平台、技术平台及互联网金融 4 大战略业务，并将云平台作为这 4 大战略业务的技术平台，全面发力云计算。

同时，还规划了京东云计算长远的发展战略——即以云计算为依托，为京东内部信息系统、外部软件开发企业和个人，以及全社会所有企业和个人，提供云计算技术平台，提供涵盖 SaaS、PaaS 和 IaaS 的云计算服务和云计算解决方案。主要业务方向包括：京东内部私有云云架构建设和维护；面向京东生态的京东开放云和京东智能硬件云业务；为全社会企业和个人提供的京东电商云，京东开发

云及行业云，等等。

京东云的战略目标是 2015 年开放京东电商业务全部资源和能力，以云的形式向全产业链各环节提供京东云服务。

京东云的发展，经历了 3 大步，第一步是京东内部各种电商资源和能力的云化，并在内部构建京东云架构；第二步是将云化后的电商资源对外开放，构建京东电商开放云平台，培育京东电商应用生态；第三步京东电商云将整合更多的外部电商资源，向全行业和全社会提供更广泛、更有价值的电商云服务。未来，京东云将面向各行各业提供全面的云计算服务，成为云服务运营商和云计算解决方案提供商。

## 内部京东云

云架构是京东正在建设的新一代技术架构，其核心是将云计算技术运用于京东技术架构各个层级，实现基础技术架构的云化升级。



京东云整体架构图



京东云整体架构是云计算能力框架的一个实现，京东云并非对云计算能力框架的完全实现，而是根据京东自己的需求，有侧重地实现。京东云架构也分为 IaaS、PaaS、SaaS 3 层。

## IaaS 层能力

京东云利用 OpenStack、KVM 等技术实现了网络虚拟化、计算虚拟化、存储虚拟化，并提供了弹性负载均衡、自动扩展等功能。

**负载均衡：**将用户请求的流量均衡地分流到各个 VM 上，不会因为某个 VM 的流量过高而宕机。它能够使你获得最大的故障容忍度，即使有 VM 宕机或者负载过高的现象，它也能无缝地为应用流量请求提供所需的负载均衡能力。负载均衡能够检测到 VM 池中不健康的 VM，然后自动地重分路由流量到健康的 VM 上，直到不健康的 VM 恢复。用户为了获得连续一致的应用程序性能，可以在单个 Zone 或多个 Zone 中使用负载均衡。

**自动扩展：**根据用户定义的条件进行资源的自动扩展或回收。使用 Policy 方式配置时，当系统达到用户所配置的需求指标峰值时，自动扩展可以保证让用户实例数量实现无缝增长。使用 Schedule 方式，用户可以实现按年、按月、按周、按天，或者按某个固定时段的实例自动扩展或回收。从而实现最大限度地降低成本，节省资源。

## PaaS 层能力

在 PaaS 层，京东云有所侧重地对 aPaaS、bPaaS、cPaaS、dPaaS、iPaaS 进行了实现。京东云在 PaaS 层提供的集成开发环境 JAE、自动编译部署、云测试、Ant 是对 aPaaS 的实现；PAF 流程平台则是对 bPaaS 的实现；缓存云、云推送等是对 cPaaS 的实现；数据云、存储云是对 dPaaS 的实现；而 MQ 消息平台、SAF 服务平台等是对 iPaaS 的实现。

### 1. 应用引擎平台

- **ANT：**为多样的 Worker/Daemon 提供一个统一的部署、配置、监控、弹性扩展的平台，以及开发框架。
- **自动部署：**自动部署系统是帮助研发运维同事上线的服务，它为大批量发布、回滚、重启应用提供了方便，能支持发布 Web 类型的项目，也能支持发布非 Web 类型的项目。自动部署的特性主要体现在以下几方面：自动部署系统规范化了上线流程，规范化了项目程序包结构，简化了上

线复杂度，缩短了上线等待时间，减少了人工频繁修改配置文件造成错误的可能性；缩短项目上线所消耗的时间和减少研发排错所用的精力，从而减少了企业消耗的人力成本，这也是自动部署系统的核心价值所在。

- **持续集成（JCI）**：持续集成是软件工程发展中保证迭代化开发质量的主要方式，通过持续集成可以利用自动化的方式来尽量自动地尽早地保证代码质量。持续集成服务提供了自定义时间构建工程，以及快速编译、部署代码包到开发、测试服务器等功能。
- **自动编译**：自动编译系统是京东项目上线流程中把源码包编译成可执行文件，最终抽取为发布包的重要一步。实现了打包编译申请、审核、配置（人工和自动化配置）、编译和抽包、进行测试、发布上线包、上线结果反馈功能。
- **自动化测试**：自动化测试平台是京东自主研发的一款测试工具。核心价值在于减少测试工作量，通过建立回归测试体系快速反馈产品中的缺陷，保证产品质量。其自主研发的测试框架Superman，适用于WebUI自动化测试并兼容流行测试框架；是一款含有任务调度、错误原因分析、数据报表视图等功能的测试工具箱。
- **移动云测试**：移动云测试平台是为移动应用开发/测试人员提供的在线应用测试平台，包括兼容性测试、UI适配、性能和稳定性测试，及自定义脚本功能测试。它可以返回包括日志和截图的详细测试报告，有助于移动应用开发的快速迭代。系统后台提供Android和iOS的真实终端设备供测试使用。
- **京东代码库**：京东代码库是基于Git、利用京东云平台存储资源的代码管理系统，包含源代码版本管理、在线编辑、Git客户端代码提交、项目信息管理、项目成员管理或派生，以及合并请求等社交编程功能。
- **Jclipse**：Jclipse是基于Eclipse 4.3.1做了二次开发的产物，包括京东工程代码生成、提供静态代码检查、代码评审快速提交等功能。它提高了开发者代码质量和规范，统一了京东的开发环境。
- **代码评审（JRB）**：代码评审是一套为京东的Code Review开发的产品，符合京东的特点，可以通过Jclipse插件或Web方式提交到评审服务器，由评审者进行review。当然，也可以多人评审，提出自己的评审意见。

## 2. 业务流程平台

- **PAF**：独立部署分布式的业务流程管理（BPM）服务。

### 3. 通用组件平台

- 云推送：客户端通过与服务端建立长连接，服务端主动实时推送消息给客户端。某应用场景为服务端往移动终端推送消息，比如给Android手机用户推送用户相关的订单信息。
- 短地址包括：长地址变短地址，实时数据分析，同时分析流量源的作用性。其应用场景包括URL地址过长的情况、移动端、邮件、短信、多媒体推广中。
- 移动分析：着眼于移动端应用全面的、实时的、专业的用户行为分析服务。云分析工具能够帮助移动应用开发商分析、挖掘用户属性和用户行为数据，以便移动应用开发者利用数据优化移动产品、调整运营模式、定制推广决策，从而提高移动产品的用户体验，获取更多和质量更高的用户。云分析支持目前主流的移动端操作系统，包括Android、IOS、Windows phone，并进行日志压缩，压缩比小于50%，单条日志小于0.3k，极大地减小对终端用户流量的影响。同时支持一键埋点，只需一行代码即可完成应用集成SDK，易于接入，开发量小。
- Cache Cloud：提供高性能、高可用、高可扩展的开放缓存平台。其应用场景为支持常见的缓存使用场景。比如，多分片、读写分离、持久化、多级复制、故障切换等。
- 日志系统：日志系统为各业务系统提供日志采集、转发、备份，订阅等服务，并提供全网海量日志检索、预警、分析服务，是企业各业务系统的日志数据总线。

### 4. 数据处理平台

- UHP：统一Hadoop平台为用户提供高可靠、高扩展、高性能、高容错的PB级别的数据存储和处理平台，并提供弹性Mapreduce计算。其应用场景包括离线、大批量的数据挖掘、转换和存储。
- CEP：基于Storm的流式计算框架，该框架提供了一套抽象、易用的类SQL语法和组件，简化常见的流式计算任务的编码工作。其应用场景包括信息流处理、连续计算、复杂事件处理。
- JProxy：提供MariaDB，MySQL等数据库的统一接入访问，拥有流量过载保护、数据自动拆分、可配置路由规则、数据无缝迁移等功能。其应用场景包括数据需要分库分表、自动扩容的应用。

- JDS: 提供一键申请、自动备份、故障自动迁移、误删数据可恢复, 以及高可靠的MySQL、MariaDB、MongoDB云服务。其应用场景为需要使用数据库服务的各种业务。
- JBUS: 准实时同/异构数据库同步, 支持跨机房数据备份。其应用场景为抽取线上数据到JDS
- JFS: 针对在线核心数据而实现的高可靠、高可用、强一致存储系统。JFS特别针对小文件进行深度优化, 支持多数据中心复制。
- JSS: 基于JFS的云存储服务, 为众多内部业务(如移动、网站交易等)和公有云客户提供可靠的RESTful Storage Service。它与AWS S3、Google Cloud Storage有着相同的应用场景。

## 5. 服务集成平台

- SAF: 分布式服务中间件, 具有FailOver、高性能调用等特性, 由RPC框架、服务配置中心组成。
- MSP: 分布式消息服务平台, 具有高可用、高性能、良好的扩展性和管理监控等特性。

## SaaS 层能力

京东云的 SaaS 主要分为自有应用系统和 JOS 开发者应用。JOS 是京东开放服务, 是面向电商的云服务平台。通过 JOS, 可以打通专家信息系统和京东的营销系统、供应链系统、物流系统、服务系统, 实现卖家信息系统和京东信息系统的无缝对接。通过 JOS, 第三方独立软件开发商(ISV)可以为卖家开发独立的商用软件系统, 并共享京东的供应链、卖家等企业客户资源。

目前, 第一步京东 IT 资源的内部云化已完成, 京东云架构基本成型, 对京东业务平稳、系统优化和效能提升的作用已经显现。电商应用关键模式实现复用, 项目平均研发周期缩短了一半以上, 大大提升了业务响应速度。实现系统弹性伸缩, 平稳应对 618 和双 11 大促波峰。服务器总体利用率提高了一倍, 网络访问量平均成本同比下降了 60%。云存储使用量突破 10P, 包括电子书、数字音乐、商品图片、历史订单等。云存储使用率超过 80%。运维自动化率达 90%以上, 系统稳定性提高, 运维人工成本降低了一半。

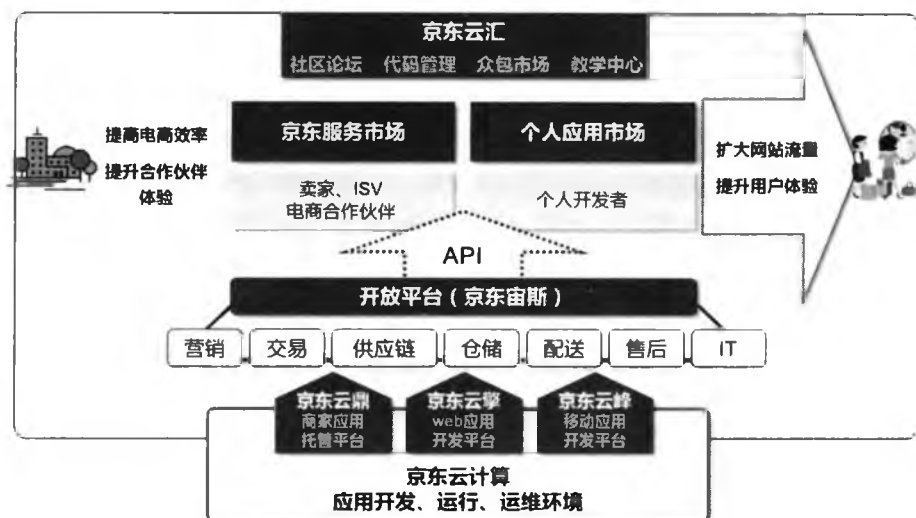
## 京东电商开放云

京东电商云是京东电商开放生态的云信息平台。为了更好地满足卖家和买家的电商应用和 IT 服务需求,京东电商云提供围绕电商应用全生命周期的云服务,与广大商家、用户、ISV 和应用开发者共同培育电商应用生态。



京东电商云生态链图

目前,基于京东电商云生态链,京东云推出了京东宙斯、京东云鼎、京东云擎、京东云峰和京东云汇 5 个解决方案,向合作 ISV 和个人开发者提供京东系统开放接口、服务交易市场、电商应用云托管平台、应用开发云平台、社区生态环境等电商云服务,初步形成了一个完整的电商云架构。



京东电商云架构图

面向京东电商开放生态上游的商家以及其他电商合作伙伴，提供的电商应用和电商 IT 服务的交易市场。在京东服务市场内，广大的 ISV、电商 IT 服务商和其他电商服务商可以将电商应用、电商代运营服务、电商培训服务、店铺装修服务等产品和服务进行展示和线上交易，实现服务商在线发布产品与服务，商家在线订购服务，服务商与京东在线结算，将好的产品和服务便捷的推广到电商上游用户手中，同时满足了上游用户的 IT 需求，提高了业务开展的效率和体验。

## 1. 京东宙斯

基于京东多年来在用户体验与自建物流等方面的电商最佳实践积累，通过营销、交易、数据、物流、售后等电商整体系统的全面开放，推出了京东宙斯（京东开放服务 Jingdong Open Service）服务。京东宙斯以开放接口的方式支持商家、供应商、软件开发商及行业合作伙伴等进行系统与业务的对接，分享京东开放成果，共建品质电商生态。并以技术开放平台为载体，结合京东云服务，将京东的“成本、效率”核心优势延伸至合作伙伴，形成整体开放生态核心竞争力。目前，京东开放服务 JOS 已对接商家四万多家，引入第三方电商应用六百多款，店铺模板三千多款，实现 8 大类总计五百多个接口的开放，全面实现有品质、高效的电商开放服务生态，并为全行业提供一站式电商服务。

## 2. 京东云鼎

京东云推出的电商应用云托管平台，为广大的商家、合作 ISV 以及其他合作伙伴提供电商应用的托管运行环境和京东电商数据开放服务。京东云鼎保证业务应用的平稳运行，实现基础资源的弹性伸缩，低成本解决资源瓶颈问题。目前，京东云鼎提供了安全、稳定、全面、可弹性伸缩的计算资源和网络环境，包括弹性计算服务、云数据库、云存储和云监控等完整的电商应用托管云服务。在京东云鼎搭建的电商应用运行环境之上，云鼎还提供订单、商品、退货、库存等数据推送服务。部署在云鼎上的电商应用可申请使用数据推送服务，实现商家应用同京东业务数据系统之间实时、无缝的数据同步，更好地满足商家业务数据实时使用和经营数据分析的需求。

## 3. 京东云擎

应用开发者提供便捷的应用开发、部署和托管服务，开发者只需将应用代码提交到京东云汇代码库，云擎自动完成应用编译、部署和托管，无须开发者进行复杂的操作和维护。京东云擎为应用开发者提供丰富的应用组件服务，对于经常使用的应用功能模块，开发者可以直接开启和调用组件服务，大大降低开发者的应用开发工作量和应用资源需求。云擎为应用开发者提供贴心的云监控、云日志等运维服务，无须关注应用的后台运营维护，通过简易的控制台就可以完成对应用的操作管理和监控分析。

## 4. 京东云峰

应用开发者提供便捷的移动应用客户端的开发服务，打造一个围绕移动应用客户端开发的全流程、闭环的云解决方案。开发者只需将应用代码提交到京东云汇代码库，京东云峰将自动完成应用的编译和打包，生成可供用户下载安装的应用客户端。在移动客户端发布之前，开发者可在云峰上进行云端测试。目前，京东云峰已经上线的云测试有上百部热门的移动终端设备可供测试选择，并提供自动化的兼容性测试、UI 适配、性能和稳定性测试及自定义脚本功能测试。此外，京东云峰为应用开发者提供了移动客户端的组件服务，对于经常使用的移动客户端应用服务，开发者可以直接开启和调用组件服务，大大降低开发者的应用开发工作量和应用资源需求。目前，京东云峰推出了移动端消息推送的云推送服务和域名网址转换的短域名服务。最后，对于上线发布后的移动应用，京东云峰还提供一款移动用户行为分析工具。云峰云分析能够帮助移动应用开发商分析、挖掘用户属性和用户行为数据，以便开发商利用数据优化移动产品、调整运营模式、定制推广决策，从而提高移动产品的用户体验，获取更多和质量更高的用户。

## 5. 京东云汇

它为京东电商云所有参与者提供的一个社区互动平台，为开发者提供交流学习、问题解答、代码托管、众包市场等服务，完成对开发者从“学习”→“生产”→“销售”的全流程支持。其中，开发者论坛围绕京东电商云以及各个解决方案、云产品进行社区交流、讨论与问题解决。教学中心为开发者、商家、京东电商云等相关方提供丰富的线上教学资料和开发经验。京东代码库实现了开发任务分包管理、代码在线和离线编辑、编码助手、代码版本管理等全程的代码管理服务。众包市场为开发者、商家、京东电商云等相关方提供互动的供需交易平台，激发平台“创富”能力。

截止 2013 年底，京东 7 大核心业务系统共开放五百多 API 接口，吸引五百五十多家合作 ISV 和三万多个人开发者加入京东电商应用生态，开发三千多款电商应用，其中有五百五十多款应用在京东服务市场交易，第三方电商应用覆盖了 80% 以上的京东 POP 商户。

# 电商云解决方案

## 京东“电商云”方案优势

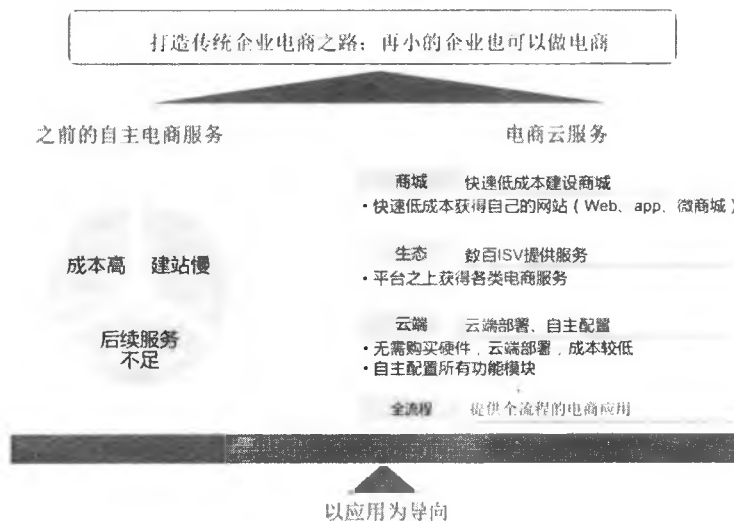
通过京东电商云解决方案，将电子商务资源进行集中化和标准化，为企业的电子商务战略带来更多的价值：

1. 提高生产力和业务价值、提高电子商务客户服务能力：通过采用京东电商云服务平台，客户可以通过使用规模小很多的 IT 投入来提供同级别甚至更高级别的业务服务水平；
2. 大幅度简化系统管理：采用电商云服务构建的 IT 运行环境使 IT 系统更集中、更简单，同时通过京东提供的自动化操作来减少潜在的人为错误；
3. 降低 IT 成本：企业可以通过购买更少的硬件设备、软件许可来降低一次性的采购成本，通过更自动化的管理降低系统管理运维人员的工作负



担，从而在整体上帮助企业降低 IT 的投入成本和运营维护成本；

4. 获得京东电商平台资源，包括但不限于：从模板装修到系统集成面面俱到的 ISV 外包服务，从流量管理到客户管理专业的电商咨询服务；

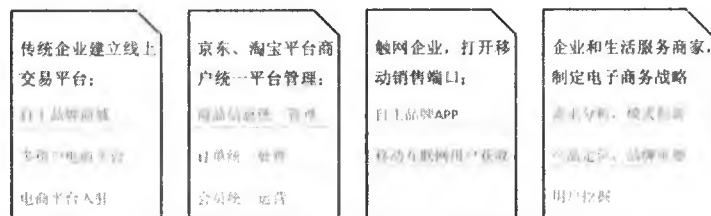


京东电商云方案优势图

## 京东“电商云”产品方案应用场景

1. 传统企业：建立线上交易平台；
2. 京东、淘宝平台商户：统一平台管理；
3. 触网企业：打开移动销售端口；
4. 企业和生活服务商家：制定电子商务战略。

京东电商云方案  
适用（主要）以下  
4个应用场景



京东电商云解决方案应用场景图

## 京东“电商云”解决方案构成

### E-Commerce

为企业提供从营销到交易平台搭建、多渠道管理集成、订单处理、仓储配送、物流优化等完整的电子商务解决方案，包括 B2B，B2C 网上交易平台。

1. 交易系统，包括但不限于 Web 交易系统、手机端交易系统（包括企业自足品牌 App 及微店）；
2. 全网营销系统，包括但不限于流量推广、社会化媒体聚合、销售数据分析等；
3. 多渠道运营系统，包括但不限于多平台统一后台管理、供应链整合、电商报表等；
4. 咨询服务。主要针对客户的特定需求和应用场景进行咨询服务，包括但不限于痛点分析、电商策略、模式创新、电商实施运营方式等，让线上业务与客户现有业务进行有效整合。

### L-Commerce

为本地零售及本地生活服务类企业提供移动互联云解决方案。

1. 微商城，为商家建立基于微信平台的线上店铺，将微信流量变现；
2. 智能商业，基于京东自主研发 Jbeacon 及 wifi 宝技术，为商家提供的本地化营销方案；
3. 大数据分析，基于 LBS 的精准营销，智能推荐及广告投放；
4. 创新服务体验，为商家提供自助定位，移动电子钱包，快捷近场支付，位置导航等服务。

## 京东智能云

京东智能云是京东云旗下针对智能硬件产品专门推出的一项云服务，致力于打造一个多方共赢的智能硬件生态链。依靠京东云强大的技术积累，我们将为合

作伙伴提供从物联网技术、大数据分析、云计算能力、开放平台、超级 App 等全方位的技术支持，从而帮助众多硬件厂家快速便捷地实现产品智能化。



京东智能云产品架构图

1. 物联网技术支持

京东智能云提供了一套开放式物联网协议，并给合作伙伴提供京东通信芯片技术和智能 wifi 芯片技术支持，从而快速便捷地实现硬件联网，并可轻松实现不同智能硬件的互联互通。

2. 大数据分析能力

京东智能云可以帮助接入平台的智能硬件进行数据收集、传输、分析和呈现，同时利用京东大数据分析能力并结合京东用户行为进行深度数据分析。

3. 多方位的云服务支持

京东智能云将把京东积累多年的云计算能力开放给合作伙伴，并能够提供弹性扩展的云主机托管能力以及海量稳定的云存储能力，同时为移动 App 提供云分析、云测试和云推送等移动云计算服务，以及为开发者提供能支持多种开发语言的弹性应用引擎。

#### 4. 构建开放平台生态

在应用层面，京东智能云将为合作伙伴提供京东信息系统开放服务、数据开放服务，从而帮助合作伙伴和第三方 ISV 开发个性化增值应用。在硬件层面，我们致力于构建一个硬件开放平台，支持合作伙伴的芯片及传感器接入京东智能云并提供给其它硬件合作伙伴。

#### 5. 强大的超级 App

京东智能云依靠完善的京东账号体系，可以让京东数亿用户使用一个超级 App 管理所有智能硬件，享受方便快捷的一站式智能生活。（作者/郭理靖、孙学斌、侯超）

# 16

## 穿普拉达的女王

### ——移动创新

#### 虚拟试镜

在移动部，创新的产品迭出不穷，比如，拍照购、颜色购、“故事”频道、手机秒杀、摇一摇等。然而，论到最创新的产品，莫过于虚拟试衣间和眼镜试戴了。

眼镜试戴项目是为了提高京东移动客户端用户体验，帮助用户了解眼镜佩戴效果的项目。它结合了增强现实技术和试戴活动运营，弥补了线上购物产品体验不足的缺席，进行了创新合理。

从眼镜试戴产品定位上来讲，我们希望所有用户都能在使用过程中，有方便、有趣的感受。因此在技术上，我们力求让每一位用户用最简单的交互，得到最好的试戴效果。

#### 项目确立

2013年，部门决定将眼镜试戴作为增强现实的第一款产品投入市场。图像识别、增强现实等技术是移动创新研发部的重要技术储备之一，结合这些技术的发展现状，在形成了初步的功能和产品形式设计后，为了保证产品的质量与效果，

创新研发部负责人 J 经理决定与国内这方面的技术专家合作，共同打造有竞争力的产品。

为此他找到了清华大学电子工程系图像图形研究所的 W 所长，W 所长带领的团队在业界有很高的知名度。J 经理这边也有从事图像处理工作多年，加上在 Sony、盛大等知名公司工作过的资深图像处理工程师老 Z 助阵。大家讨论了需求后，基本敲定了产品的技术指标和攻坚难点，接下来双方又进行了细致深入地调研，对整个项目的工作量有了宏观的估计。在 J 经理的推动下，京东与清华大学签订了眼镜试戴项目的合作合同，一场移动互联网工程师与顶尖大学科研人员的思维碰撞就此拉开了帷幕……

## 第一个山头：PC Demo

“检测、跟踪、投影、绘制”，这 4 个词几乎异口同声地从老 Z 和清华大学负责眼镜试戴算法工作的 Z 博士口中说出，“行家一出手，便知有没有”，双方对于整个项目的工作模块划分有着高度的认同。接下来 Z 博士研究了多篇论文提出了几种技术方案，老 Z 凭借多年的实际项目开发经验，从工程和维护等角度进行了可行性的论证，京东方面进行 App 开发的同事也从算法移植的角度给出了意见反馈，最终大家选定了方案，并先从 PC 版的原型开始做起。

不到一个月的时间，第一版的 PC Demo 就做出来了，大家兴奋地在笔记本的摄像头前试戴电脑中的 3D 眼镜：有的“搔首弄耳”摆出各种姿势引来大家围观、有的“摇头晃脑”配合夸张的表情检测眼镜能跟踪的范围……这个 Demo 在移动部的员工间也引起了小小的轰动，创新研发部以外的同事也纷纷来体验：“有意思”、“牛”，这是大家对产品的第一印象，大家的认可也验证了方案的可行性。之后老 Z 深入分析了算法代码，提出了几个可以提高检测精度和范围的改进建议，PC 版的 Demo 又经过了几轮优化更新，至此，眼睛试戴在 PC 上基本达到了大家预期的效果，算法最终也被确定下来。

## 第二个山头：移动 iPad

PC 版的 Demo 成功，只能算是整个项目的可行性验证，真正实现手机上的眼镜试戴还有许多工作要做。首先面临的问题，就是手机处理器的性能比 PC 要低很多，计算复杂度太高的算法在手机上运行速度要比在 PC 上慢很多。为了降低开发风险，J 经理首先安排了 iOS 开发组的两位美女，D 姐和小 C，来负责将算法移植到 iOS 设备上的工作。为什么安排两位美女就可以降低开发的风险呢？因为在 J 经理的部门中，女生都集中在 iOS 团队，安卓团队全是男生。由于安卓

的机型多种多样，屏幕尺寸、机器性能参差不齐，因此在安卓设备上开发需要验证各种机型下的效果，工作量成倍增加，无疑会推迟产品的最终上线时间。选择 iOS 设备，还因为 Xcode（iOS 设备的开发环境）支持 C 语言直接编译，从而降低了移植算法的工作量，因此这一决策为项目的如期上线提供了保证、降低了风险。

D 姐不仅有丰富的 IOS 开发经验，还有图像处理的工作背景，对 OpenGL 三维绘制技术也做过一定的研究，她作为 App 开发的负责人再合适不过。D 姐用了一周左右的时间就将 PC 上的算法在 iOS 设备上运行了起来，小 C 帮助 D 姐实现了测试 App 的交互界面。于是大家开始献出各自的苹果设备验证算法的性能：“iPhone4 好慢啊，画面卡住了不能动弹！”、“iPad 上效果不错，我这个是 iPad2，谁有 iPad1？什么？iPad1 没有摄像头？！那算了-\_-！”、“看我的 iPhone5 多牛，运行起来好流畅啊！”……通过实验，确定了性能可以支持算法的设备是 iPhone4s、iPad2 及其后续的苹果机型，J 经理下令先开发 iPad 版，在用户量较小的 iPad 版上收集用户反馈后，进一步优化产品和性能再推出 iPhone 版。

### 迁移的难题

然而移动部的同事们，面临着残酷的现实：从零开始到发布，只有几个月的时间，怎么办？使用时，用户的脸型、眼睛位置、光照条件各种各样，怎么办？试戴的时候会拿着手机、平板左右晃头来看效果，戴歪了怎么办？移动设备处理能力太弱，跑不动复杂的佩戴算法怎么办？没有解决不了的问题！兵来将挡，水来土掩。

**时间紧：**我们团队联合清华大学电子工程系图像图形研究所共同进行算法开发，利用著名高校长时间的技术积累，加快算法的开发进度。

**算法难：**在反复地对高校积累的各种算法及对当前业界领域内最好的相关产品进行剖析、研究之后，我们团队吸纳百家所长，加以改进，经过反复无数次的尝试，终于确定了超过业界顶尖水平的虚拟眼镜佩戴算法。通过加大各项适应能力以及不断与用户初始化时的状态进行比较，我们的产品最大化地减小了用户的脸型、眼睛位置差异以及试戴时的设备抖动对最终佩戴结果的影响。并且用户只需在应用开始时点击屏幕一次，即可完成佩戴或纠正戴歪的眼镜。

**优化艰：**算法确定之后，团队同事很快地将佩戴模块移植到了移动设备上，这时新的问题产生了：慢！非常慢！由于硬件架构差异巨大，算法在移动设备上的处理速度比测试平台慢近百倍！于是，大到整体算法流程，小到每一个变量的声明调用，每一行代码如何更好地适应移动设备的硬件特性，都被团队同事反复

推敲优化了若干遍。模块需要用到的所有函数及外部库都被彻底分析、调整、优化。

## 第二座山头拿下

在进行算法移植的同时，移动部产品部的各路好汉也开始加入眼镜试戴项目，进行 iOS 设备上的产品设计、交互操作、模型制作和美工切图等工作。首先出马的是 L 大师产品部为数不多的 T4 级产品经理。大师话不多，先是听大家的想法、看各种竞品，之后一份厚达三十多页的 PRD 重重地落在研发同事的手中！大家不仅没有被大师的气势吓倒，反而边读边讨论，大师则耐心给大家讲解产品设计的细节，虚心接受大家的建议，PRD 改了又改，原型图画了又画，还让负责用户调研的 X 同学给同事试用 Demo 获取意见，最后产品设计定型。

D 姐和小 C 快速按照 PRD 进行客户端的开发，同时项目负责人 H 和老 Z 对算法又进行了深入地优化，从工程上和核心指令上都进行了改写，使用到了 Arm Neon 技术在代码底层利用特殊指令来优化速度，取得了近 30% 的运算速度提升。产品部的小 W 根据大师的选择，制作了 20 副 3D 眼镜模型，其中既有白领学生的近视镜，又有时尚潮人的太阳镜，还有复古卡通的趣味眼镜，可以满足各类用户的试戴体验。经过了设计、开发和测试，iPad 版的眼镜试戴功能于 2013 年底正式发布，用户进入京东商城 App，点击试戴按钮，将摄像头对准自己，就可以在屏幕上看到自己佩戴 3D 眼镜的效果了……

## 第三座山头：iPhone 端，不仅仅是迁移

iPad 版的上线，鼓舞了团队的士气，经过一段时间的用户反馈收集，iPhone 版的开发于 2014 年春节后正式发布。这次的产品中还增加了“照片墙”的功能，产品部的实习生小 D 领衔照片墙的产品设计。

这位 90 后帅小伙，更懂得时下年轻用户的产品需求，他分析现在的用户往往不满足于产品的功能，他们还要参与、分享、秀出自我。“照片墙”就是为了满足他们试戴眼镜后与别人分享、展示给所有人浏览的需求，同时，浏览者还可以给喜欢的照片“点赞”。另外，眼镜试戴分作多期活动来运营，每期主打不同的眼镜主题，并为活动中得“赞”最多的用户赠送实体眼镜。

这款产品在第一期活动中就得到了两万多张照片分享，效果超出了预期。值得一提的是，在 iPhone 版眼镜试戴的开发过程中，通过之前团队的磨合，项目负责人 H 将敏捷开发模式贯彻得彻底而高效：春节后仅用了 6 周时间就完成了开发，并在 2014 年 4 月发布。两个迭代各分 3 周，第一个迭代做了数据库与接口设计、



假数据联调、服务端和客户端的代码框架开发，第二个迭代做了服务端功能逻辑完善和客户端界面联调，测试并修改了 Bug，最后还增加了数据统计需要的埋点且如期上线。整个团队协同作战，高效地完成了 iPhone 版的开发，为部门和团队积累了宝贵的项目和开发经验。

看着“照片墙”上用户的一张张笑脸，J 经理也露出了满意的笑容，这款产品正是为这些用户足不出户就能佩戴各种眼镜而设计，最终也达到了这样的效果。2014 年 5 月 13 日至 18 日，眼睛试戴作为京东创新类产品的代表之一，参加了北京市第十七届科技产业博览会。展会上北京市的领导莅临了京东展位，肯定了京东创新方面的工作。来到京东展台前的更多的是普通观众们，在第一次试用了眼镜试戴后都发出了由衷的赞叹……

配合运营同事的相关市场活动，我们上线的虚拟眼镜试戴模块取得了巨大成功。同时预料之外地，用户开发出了新玩法：由于佩戴算法的适应性强，一些用户甚至将刚出生的小宝宝，自己养的猫、狗等宠物戴上了我们的虚拟眼镜，分享在各个社区平台上。

## 虚拟试衣间

### 用户愿景

2014 年 1 月的一天，京东用户丽丽像往常一样用 iPad 打开京东购物客户端，她准备挑选一件漂亮的羽绒服，和往常不一样的是这次京东客户端弹出了一个醒目的提示窗口，窗口中文字的大意是京东购物客户端发布了一个重要的新功能——京东虚拟试衣间，敬请用户试用。



京东试衣间

丽丽其实早就希望京东有这样的功能了，因此非常想试一下，于是她点了“确定”按钮，不一会儿，iPad 提示新版的京东客户端下载完成，是否确定安装，丽丽迫不及待地又点了“确定”键。新版京东客户端很快呈现在丽丽眼前。

新版客户端界面和以前的版本大致相同，不同的地方是中间的工具栏上多出了一个漂亮的图标，这个图标是一件漫画版的衣服，下面写着一个“穿”字。

丽丽怀着激动的心情马上点了一下这个新图标，几秒钟后，一个崭新的界面呈现出来，左侧是一个身上只穿了内衣的漂亮女模特，右侧是多列各式各样的衣服，通过简单的用户使用说明丽丽了解到，这个新发布的京东虚拟试衣间可以让用户用虚拟试衣模特搭配试穿京东在售的自己感兴趣的衣服，可以非常形象地看到自己感兴趣的衣服在模特身上搭配出的效果，丽丽兴致勃勃地将右侧列出的衣服用手指拖到左侧的虚拟模特身上。只这么一拖，衣服眨眼间就整整齐齐地穿好了，而且里外各层衣服会自然地搭配在一起，比如，衬衣穿在内衣外面，毛衣穿在衬衣外面，而不会反过来，不管丽丽先穿衬衣还是内衣，用起来非常方便。相同类型的衣服也不会重叠，而是自动替换，比如，模特已经穿了一件衬衣，丽丽再往模特身上穿另外一件衬衣时，模特身上的衬衣会自动脱掉换成新的。丽丽不用费什么脑筋就可以尽情地往模特身上穿各种衣服，方便极了。当然脱衣服就更方便了，用手指将模特身上的衣服往外一拽就可以了，（当然，很明显，模特最里层的内衣是不能被拽掉的）。



模特试穿图

丽丽兴致勃勃地“玩”了半个小时，给模特搭配了各种衣服，找到了好几种自己特别满意的搭配效果，这时丽丽突然想到了一个问题：左侧的虚拟模特确实是非常漂亮，身材也很棒，但这毕竟不是丽丽自己啊，衣服穿在别的模特身上再漂亮也并不代表穿在自己身上会很漂亮啊。丽丽仔细比较了系统提供的模特和自己的身材、肤色、发型，发现自己比这个模特要稍微胖一些，肤色也不太一样，头发的颜色和长度也有较大差异。丽丽开始怀疑衣服穿在自己身上和眼前的这个漂亮的虚拟模特身上效果会有较大不同。丽丽的怀疑非常有必要，这也是京东推出虚拟试衣间的目的——让用户自己做模特！

丽丽想，京东既然推出了虚拟试衣间，不会光让用户用系统推荐的、身材完美的虚拟模特试穿吧，肯定还有别的功能。她仔细看了试衣间下面的工具栏，发现有一个写着“用户拍照”的按钮，为什么要让用户拍照呢？这里肯定藏着新的功能，丽丽按了一下按钮，又出现了一个提示窗口，内容让丽丽既期待又激动：用户可以用自己的头像照片和身高三围数据自创虚拟试衣模特，果然是这样！丽丽兴奋不已，立即按照提示给自己拍摄了一个正面的头像，可爱的京东小 Joy 出现了，提示丽丽“它”正在努力工作，帮丽丽建立个性化的试衣模特。短暂的等待后，小 Joy 提示的“它”已成功地识别出了照片中丽丽头像的确切位置，并用

红线标出了头像的准确边缘，而且还识别出丽丽是女生（如果用户对识别的准确性表示满意可以进入下一步，如果不满意，可以让小 Joy 重新努力一下或者用户重新拍一下照片）。

由于丽丽拍的照片很清晰，光线也充足，小 Joy 很轻松地识别准确了，丽丽很满意，点了下一步，又一个新的界面出现了，左侧是丽丽的虚拟试衣模特预览图，右侧是丽丽的头像，头像边缘标着红线，红线上还有一些小圆点，通过移动这些小圆点还可以对虚拟试衣模特的效果进行微调，以便于达到更加精确的效果。

京东试衣间帮丽丽配上了一个肤色和丽丽头像一致的虚拟身体，两者非常自然地结合在一起，完全看不出是拼接上去的，丽丽稍微调节了一下头像大小后又进入了下一步，界面呈现的图像是一个完整的虚拟试衣模特，头像是丽丽的，所以看上去和丽丽长得一样，但身材还不太一样，因为到这一步之前丽丽并没有告诉京东试衣间她的身材是如何的，而这一步的功能正是设置模特身材。



换脸效果图

丽丽发现这一步做起来非常容易，虚拟试衣模特旁边出现了一些滑动条，只要用手指轻轻左右拖动这些滑动条，模特的身材就会立刻发生变化，如，拖动肩膀附近的滑动条肩宽便发生变化，有一种所想即所得的感觉，这样的滑动条共有 12 个，可以全面地控制虚拟模特的身材。



设置模特身材图

丽丽输入自己的身高和体重数据，滑动了控制三围的滑动条后，丽丽觉得现在的虚拟试衣模特无论从长相、身材、还是肤色来说都已经很像自己了，这回可以用这个和丽丽一样的虚拟模特（京东试衣间称为个性化虚拟模特）帮丽丽试衣了。

因为模特和丽丽很像，丽丽一下子认真起来，刚才还觉得好像在玩一个游戏，现在觉得不再是玩游戏了，而是认真地给自己挑选起衣服来。丽丽看到了一件紫色的羽绒服，平时丽丽一般穿红色或橙色的，从来没试过紫色的、既然现在试衣服这么方便，那就各种颜色，各种搭配都试一试吧，穿上紫色的羽绒服一看，丽丽发现紫色和自己的肤色还挺配的。她这几年有点发胖，原来的红色和橙色穿起来看上去有点显胖，换上紫色一看，还挺显瘦的，丽丽心里美滋滋的，决定这项换个风格，买件紫色的。

等等，先别下单，丽丽发现京东试衣间还有一个重要功能——分享，就是说，丽丽穿上紫色羽绒服的样子可以马上让丽丽的朋友们看到，看看朋友们是怎么评论的。丽丽按了一下分享按钮，京东试衣间告诉丽丽，她可以将试衣效果图分享到微信朋友圈、新浪微博等热门的社交网络，朋友圈里密友多，丽丽将效果图分享了出去，还特地加了一句想跟朋友们说的话。

微信朋友圈的反馈很快，丽丽的朋友们马上发现了“丽丽的新装”，纷纷写评论，“点赞”，丽丽看着满屏朋友们的赞美和询问，心里甭提有多满足了，心里想：哈哈，你们还不知道京东有虚拟试衣间了吧，这回 out 了吧。很快，在丽丽

的介绍下，朋友们也知道京东试衣间了，纷纷进来尝鲜，试衣间里越来越热闹，朋友圈里也越来越热闹。

## 研发方案确立

在“丽丽”们使用京东试衣间的同时，京东试衣间的开发团队正在密切地关注着服务器里显示的用户数据，看着后台数据的快速增长，最激动的要数京东试衣间项目的负责人老 G 了，他已经为这个项目默默工作了 6 年，在前 5 年里，老 G 一直在埋头研究虚拟试衣的解决方案，其实虚拟试衣的概念已经在网间流传了十几年，德国的弗劳恩霍夫学会的科学家们最初用 3D 扫描仪扫描真人的身体建立起真人的 3D 虚拟模型，这种方法在实验室里是可行的，但在商业领域却不可行，因为商业领域的可行性基于效果和成本的平衡，用 3D 扫描仪的方式对用户来说显然过于“高级”，3D 扫描仪的价格不菲（目前，民用级的 3D 扫描仪价格在 10 万量级），操作也并不简单，大部分用户不是“工程师”，是没有办法在家里操作 3D 扫描仪的，高昂的成本也是用户和商家难以接受的。

6 年前，摆在老 G 面前的难题是要开发出一套商用的个性化虚拟试衣系统。商用就是要在试衣效果（个性化程度），用户操作难度（易用性），运营成本（虚拟试衣模特与服装的制作）这三者之间做好平衡。3D 方式成本太高，老 G 很快就放弃了这个解决方案，2D 方式技术落后，视觉效果差，也很快被排除。老 G 最后定下的方案被称为 Z3D 方式，就是说基于 3D 的内核技术，但只展现正面的视觉效果（也可以延伸到背面，侧面），虚拟试衣模特和服装建模只针对正面（背面，侧面），这样就大幅节省了成本，完全可以基于用户和服装的照片自动建模，同时 3D 技术的光影效果可以带来视觉上的立体感和真实感，效果大幅优于 2D 的效果。

方案听起来虽然比较清晰，但仍然有很多难题需要解决。首先，方案不能对用户有较高的操作复杂度要求，老 G 的想法是，用户只需要给自己拍一张正面的头像，然后告诉计算机身高和体重，计算机要能够自动建立一个和用户相似的虚拟模特。要实现这样的想法需要计算机具有一定的智能度：首先，计算机要能够认出用户照片中的头像（这一点在目前来说已经不是很难，人脸识别技术发展很快）；其次，计算机要能够准确地将用户的头像从照片中提取出来，这个步骤需要准确地区分出用户头像和背景的边界；然后，计算机要确定用户的肤色，给出一个合理的平均值，因为需要将用户的头像和计算机事先准备的虚拟模特身体完美的结合在一起，不能出现肤色明显不匹配的情况；接下来，还要让虚拟模特身材可以自由地调节，像捏橡皮泥人一样改变身材，要实现这个功能需要给虚拟模

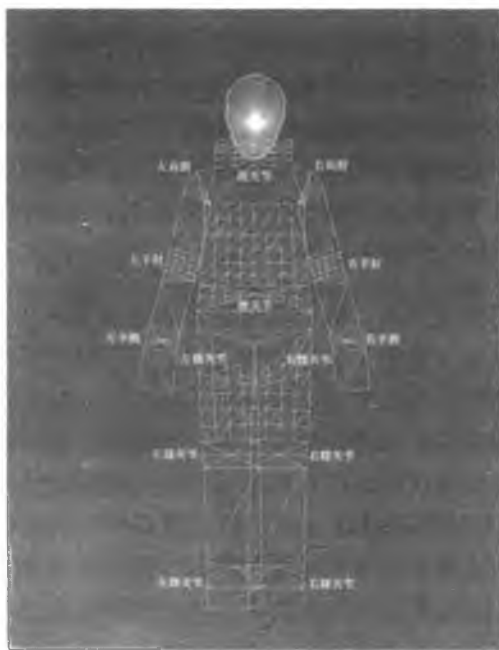
特的身体建立起一套像人体一样的骨骼皮肤系统，骨骼可以运动，皮肤与骨骼距离可调节，从而实现姿势和胖瘦的改变。

## 主要问题解决

经过了多年的研究，老 G 终于找到了一个比较合理的骨架结构，有了这个结构，虚拟模特的身体真的可以很自然的减肥和发胖了，肤色也能自由调节，还能做出各种试衣的动作，而用户使用时仅仅需要拍个照片，移动几下手指即可。

接下来的难题是如何给虚拟模特“穿上”衣服，如果虚拟模特的身材和姿势是固定的，那么这个问题将变得非常简单，但显然这样是不符合预期的。同样一件衣服需要能够穿在不同身材，不同姿势的虚拟模特身上，看上去还要比较自然合理，不能只是一种拼凑的效果。

给不同的身材、不同姿势的虚拟模特制作不同的衣服（也就是说，同一件衣服要建模很多个版本）的问题是能够得到解决的，但这一做法违反了商业化系统中的低成本原则，每件衣服需要制作多个版本，而网上的服装数量又十分巨大，这将意味着巨额的服装制作成本，用什么方法解决这个难题呢？老 G 为此费尽心思，设计了很多种方案均被他自己一一否决。



骨架结构图

一天，一筹莫展的老 G 在奥林匹克广场中一边走路一边思考，突然一个很像一件衣服的风筝在广场中被放了起来，风筝由细细的骨架和薄薄的蒙纸做成老 G 看到了这个风筝，突然受到了启发，风筝的结构其实就是衣服的结构，可以给衣服设置一些虚拟的骨架，将衣服撑开，骨架的大小可以改变衣服的尺寸，骨架的运动可以牵动衣服的运动，从而改变衣服的姿势。只要将衣服的骨架和虚拟模特的骨架对应起来，衣服就能跟随虚拟模特的身材和姿势发生相应的改变。这个想法很快被老 G 用 C++ 语言实现在软件中，经过长时间的优化和改进，老 G 终于对呈现的效果表示满意。



衣服骨架图

## 上线

2012 年 10 月，Z3D 虚拟试衣系统 1.0 版终于诞生了，老 G 为其申请了软件专利。并在京东迅速建立起一个虚拟试衣项目组，开始研发基于 Z3D 的京东试衣间，基于 Windows 版的虚拟试衣软件很快被移植到 iOS 版本，并进行了多处改进，不到一年的时间里，虚拟试衣组在老 G 的带领下，推出了京东试衣间 1.0 版，该版本移植了 Z3D 的大部分功能可运行在 iPad 平板电脑上，软件内嵌于京东购



物客户端中，用户无须下载别的客户端（App）即可体验京东试衣间。

市场和媒体对京东试衣间的反应很积极，大量主流媒体开始报道京东试衣间，甚至有爱好者开始给京东试衣间撰写攻略——“带你玩转京东试衣间”，京东试衣间也出现在各种展览会中。这项行业领先的技术开始与电商平台紧密结合在了一起。（作者/李松峰、赵刚、靳简明、候会满、张斯聪）

# 17

## 少数派报告——大数据洞察未来

### 京东大数据平台演变历程

大数据，时下已成为人们讨论最多的热词之一。同样，在京东，大数据也受到越来越多的关注。

2013 年年初，在京东制定的下一个 10 年发展规划中明确提出，大数据战略将成为公司重点战略方向之一。从此，大数据正式冠以战略高度登上京东舞台。其实早在这一战略计划宣布之前，京东大数据的应用价值已经体现在公司业务的各个方面。如：用户消费行为深度挖掘、EDM 精准营销策略实施、销量预测与库房自动补货、搜索推荐系统的持续优化，广告精准投放技术等，这一系列对公司运营至关重要的项目背后，都是对大数据创新应用的不断探索与研究。伴随着京东业务的调整与发展，高价值的业务数据积累与大数据技术的演进，京东大数据创新应用与产品化进程不断取得新突破。大数据挖掘平台、实时大数据解决方案、推荐搜索系统、电子商务全链条企业级数据仓库，大数据在京东发展的道路上背负起更加重要的使命。

在行政归属上，京东大数据平台直接隶属于京东集团，作为基础数据技术平

台，面向京东商城、拍拍、京东金融及海外事业部提供数据服务，并承担部分对外数据服务的职能。平台致力于电子商务海量数据处理技术研发与应用，打造高性能、高稳定性、高安全性的数据治理、数据分析、数据挖掘基础平台，为京东及电子商务行业挖掘大数据价值提供全过程解决方案及技术保证。自主设计研发的企业级数据仓库历经 618、双 11 等高并发的数据生产性能考验，支撑着京东的决策数据分析与高附加值的数据应用产品。

京东大数据平台从无到有，从集中式到如今的分布式，从 Oracle 数据仓库到 JDW2.0，4 年期间完成了复杂的演变。Tiger，作为京东大数据平台负责人，也是大数据平台的总架构师，伴随着京东数据仓库发展，他一直在思索两个问题：如何建设电商特有的复杂业务的数据仓库？如何在保障安全的情况下降低使用数据的成本？打造一个“人人皆可成为大数据专家”的“自助式商业智能（BI）平台”似乎是唯一出路。但如何实现呢？

事情还要追溯到 2009 年 12 月的一天，当邢捕头和其他 3 位同事在苏州街的会议室里参加数据部成立准备会的时候，作为新部门的第一位经理，他不无惆怅的想，采销的扔过来那么多数据需求，就他这几个人可怎么办！但他几乎没有任何时间感叹了，公司订单增长这么快，放眼望去，各部门的同事都像打仗一样从早到晚忙个不停，采销“讨账”的同事马上就要上门了。过去几个月来，对数据各方面需求的暴涨让技术研发体系的老大老李意识到了数据的重要性，支持业务部门数据需求的同时也有必要为将来做规划，于是便准备成立这样一个部门。半年后 Tiger 作为数据部总监加入京东，数据部正式宣布成立。

2014 年 8 月一个晴朗的早晨，当邢捕头作为特邀讲师站在清华大学电商总裁特训班的讲台上时，早已惆怅不再，但激动依旧。因为他有足够的底气面对台下几十名电商从业精英论道电商大数据。作为京东大数据平台数据创新部的高级经理，他所负责团队几经变换之后已经扩大了近 10 倍，大部分成员为高级数据挖掘工程和高级数据分析师，肩负京东大数据创新应用的重要职责。而 Tiger 带领整个大数据平台此时已经达到 200 人的规模，除了邢捕头的数据创新部外，还包括数据架构部、数据产品部、平台运营管理等多个部门。

光明的前途往往都要经过曲折的道路才能抵达，今天的数据部也同样经历了不少的曲折发展历程。2010 年下半年，数据部依据服务职能对象的不同拆分为两个部门。Tiger 团队主要面向采销、市场等部门提供数据支持，另一个数据团队主要面向仓储、物流等部门提供服务。

Tiger 团队当时的工作重点是数据分析与数据报表，但在没有数据仓库的年

代里，分析师们只能依靠 DBA 提供的两台查询机排队写 SQL 提数，然后再拿回来做分析。遇到业务需求方描述不清或口径不明确时，往往要花大量时间沟通调研，反复提取数据，这个过程耗时费力不说，还会被需求方不断催促。不能自主的数据提取过程让数据分析与使用的效率非常低下，尽管分析师没日没夜的加班，但还是不能满足采销业务的需求。当然业务人员也无法理解分析师们的难处，有次甚至因为大量明细数据未能及时提取把当时的分析师玉兰姑娘逼哭了。高速成长的京东，对身处其中的每一个人也都是一场难忘的历练，偶尔大家凑在一起聊天，不无自嘲的感叹：在京东，女人被当男人使，男人被当牲口使。哈哈一笑之后，大家也都在心中庆幸能有这样的机会快速成长，不仅在工作方面学到很多，更重要的是成长的收获。

数据服务不能自主的困境，Tiger 看在眼里，急在心里。同时，京东业务一直在高速发展，订单量及交易额屡创新高，商品品类在不断丰富，新开仓库在全国各地陆续启用，以前简单粗暴的数据提取与分析很快会成为瓶颈。因此，长远来看，自建数据仓库是做到自主数据服务的必由之路。不仅如此，Tiger 做了更长远的打算：数据服务不仅要自主，而且要做到数据需求方自助服务！自建数据仓库的同时，自主开发配套的数据调度生产、数据分析提取、数据知识管理、数据报表呈现及数据质量监控的产品体系，并建设相应的数据集市联合推广使用，这，其实就是今天京东的大数据平台。

与此同时，另一个数据团队几乎也遇到了同样的问题，自建数据仓库也成了他们的不二法门。但与 Tiger 团队最大的不同在于，从电信等行业加盟的决策者认为借鉴传统行业数据仓库的成熟经验，购买 Oracle 等服务商提供的商业产品会更快更有保障。短期来看，确实如此，两台 Oracle RAC 小型机部署之后，数据处理效率明显提高。造价不菲的 Oracle BIEE 商业智能平台在数据处理与展现方面显示出强大的威力。而 Tiger 团队自建数据仓库的初始版本基于开源关系型数据库管理系统 MySQL，采用集群（Cluster）应用架构，支持大规模数据存储与计算应用。数据提取与报表展现产品同样采用开源技术自主研发，后台 Java 程序，前端采用 Ajax 框架 Extjs。这种轻量级的数据产品在稳定性及服务能力上与 BIEE 相比虽然有一定差距，但支持快速迭代及无与伦比的灵活性这些优点也是成熟商业产品所无法替代的。正是自主研发、快速迭代和灵活应用这些适用于互联网企业的特点，使得一系列产品和技术方案在快速成长的京东很好地支持了各种场景的数据需求，也成就了今天的京东大数据平台。这对于其他成长中的互联网企业也具有重要的借鉴意义。电信、银行等传统企业的数据仓库解决方案以其稳定性著称，但前提是这些传统行业的业务体系也较为固定，数据库、表结构不会轻易

发生变化。而互联网企业业务变化频繁，不但要支持横向扩展，甚至在整体业务结构层面都面临着平均每半年一次的调整。因此，随着时间的推移，撇去成本不谈，成熟商业产品的使用局限性也越来越明显。

2012 年年初，京东研发体系再次调整，数据部又重归于一体，之后的大数据平台部由 Tiger 统一负责。此时的 Tiger 团队已开始着手进行分布式数据仓库的建设工作，今天的京东企业级数据仓库 JDW 便是在当初设计的版本上演变而来。与此同时，打造一个“人人皆可成为大数据专家”的“自助式商业智能（BI）平台”的蓝图也逐渐清晰。通过数据分析工具，一个不懂数据仓库的数据分析师也可以完全根据自己的需求，随时随地获取数据、分析数据，为业务运营提供决策支持。

基于 Hadoop，以 Mapreduce 作为计算引擎的分布式数据仓库可以说是大数据处理的“标配”，2012 年 8 月，由 40 台机器搭建的第一版集群上线，相较于 Oracle 小型机，性能明显提升，让我们第一次体会到分布式的威力。2012 年 12 月，集群服务器数量增加到 110 台，但由于带宽限制，性能遇到瓶颈，于是在 2013 年 3 月升级万兆带宽，同时服务器数量增加到 200 台。3 个月后的 2013 年 6 月，MySQL 数据仓库完成迁移，正式退出京东主数据仓库，集群服务器数量达到 310 台。2013 年 11 月，京东全业务线数据接入集群，可满足公司各体系业务，服务器数量达到 440 台。2013 年 12 月，老葛、彦明带领一众 BI 兄弟连续奋战三个月之后，终于完成 Oracle 数据仓库迁移，相关任务全部下线，Oracle 正式退出京东数据仓库。2014 年 3 月，集群服务器数量达到 800 台，仓库模型全面升级到 JDW2.0。调度平台功能逐渐完善，生产模式也由原来的单节点调度升级到分布式调度。数据开发、元数据管理、数据质量监控、数据展现及统一权限管理产品全面推广应用。基础数据经过数据仓库加工后通过数据集市的方式面向公司各体系开放服务，同时，面向集市用户的产品运营与认证培训工作建立标准体系，至此，京东大数据平台逐步走向成熟。

2014 年年中的时候，公司整体组织调整架构，研发部门纳入相应业务体系垂直化管理，大数据平台依旧归属京东集团，面向京东商城、京东金融集团、拍拍及海外事业部提供大数据应用与技术服务。新的起点，新的期待，部门组织了一次集体出游，在苍茫辽阔的大草原展现团队恢弘气势，为更高更远的目标凝聚力量。



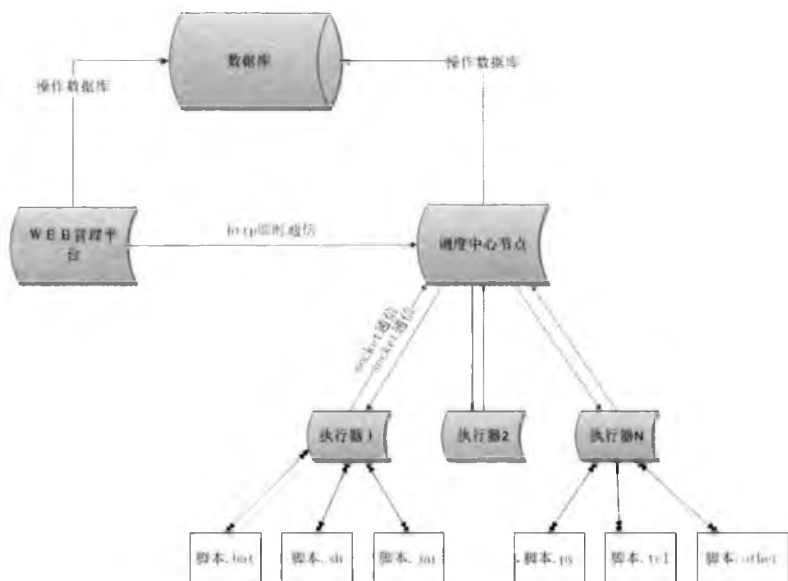
大数据平台团队合影

## 京东大数据产品

对于刚刚成长起来的京东大数据平台来说，数据产品并不是一个新鲜事物，2011 年自建数据仓库上线的同时，第一款数据产品调度平台也一同上线并正式投入使用。在 Tiger 的指导下，调度平台无论是 UI、功能还是用户体验各方面都获得一致的好评。

### 调度平台

订单交易，仓储物流等众多京东系统都会产生数据，仅日志内容每天的大小约为 1TB，大量的数据如何统一汇总到数据仓库来呢？这就需要调度产品来实现数据生产。京东调度平台发展至今已经是 3.0 版本，每一次的更新迭代都凝聚着阿敏等几位开发工程师的许许多多多个日夜的心血，也是我们技术突破与功能升级的具体体现。



调度平台 1.0 版本架构

1.0 版本于 2011 年 8 月上线，一台服务器作为中心节点指挥调度，另外 3 台服务器负责相关数据作业，任务之间通过后置变量的方式设定前后依赖关系，调度机制便运行起来了。数据仓库建立之初的任务并不太多，数据量没有太过庞大，数据 ETL 过程所需计算资源也都完全应付得来。但随着仓库收纳数据的增加，数据生产任务越来越多，任务之间的依赖关系也变得越来越复杂。每个 BI 工程师需要根据自己的生产任务设定后置变量的值以建立任务依赖关系，任务多了之后不但设置起来耗时费力且不易管理，当一个人的任务需要重跑时后置变量的修改可能会影响到别人的任务。

2.0 版本上线了新的调度引擎，彻底解决了这个问题。新任务上线只需要选择依赖的父任务即可建立关系，且流程独立，不会因同一个任务被多个依赖而造成干扰。除此之外，任务可视化配置与浏览功能也在这个版本上线，任务运行状态监控预警功能也投入使用。这之后的功能升级也一直在进行，较大的功能改进莫过于虚拟节点了。数据生产过程中，尽管几率很低，但仍然还是会出现一些物理节点挂掉的情况，而这种情况一旦出现，影响将会很严重。于是，虚拟节点的功能应运而生，原理就是在原来的物理机集群上做一层虚拟化，如果遇到生产节点故障的情况自动切换到另一个节点。同时，根据不同节点的负荷，将新的任务自动分配到负荷较小的节点，做到负载均衡。这一系列功能的上线使得平台的稳定性大大提高。

3.0 版本从功能上更加丰富，并且实现了数据生产的半自动化运行机制。所谓的半自动化是指数据任务可在配置目标数据库、表之后自动生成 ETL 模板并完成数据清洗，之后是人工创建调度任务完成数据生产。另外，自主研发的抽数模块 Plumber 也在这个版本中上线，Plumber 技术实现了异构数据库之间的快速数据交换，且具有较高的稳定性，数据导入导出的维护成本也大大降低。还有服务器运行状态的监控系统 Phenix 也集成到了调度监控中，实时采集服务器运行状态数据并对服务器心跳、存储空间使用、CPU 资源消耗等进行预警。对于上游系统故障造成的历史数据补充问题，之前的版本中需要人工查找相关依赖任务，然后一个一个配置参数后点击重跑，而在新的版本中 BI 工程师们期待已久的一键重跑功能上线了，并且支持批量操作一千多个重跑任务，BI 工程师加班到半夜一个点鼠标的日子永远成为了过去。

作为大数据平台的核心系统，调度平台不仅承担着数据生产的重要使命，同时也负责集市数据推送、模型数据加工等任务，部门有超过 1/3 的人都在围着它转，重要性可想而知。今后的功能升级迭代将在增强生产能力的同时更加注重自动化服务及开放运营等平台产品的特性，为大数据管理及挖掘大数据价值提供可靠保障。

## 数据集成开发平台

数据集成开发平台是京东大数据发展的一个里程碑产品，它的出现结束了数据分析师和业务部门数据需求人员通过客户端工具手工提取数据的痛苦经历，并对后来的数据知识管理平台等产品的出现产生直接影响。当前平台用户接近 1000 人，数据订阅任务总量逾 40000 个。

早期版本的数据集成开发平台就命名为提数工具，提数也是当时寄予这款产品的最重要的期望。相信每一家公司对于数据的需求都是“刚需”，快速发展的京东，流动并且快速流动的数据更是像一个人身体里的血液一样不可或缺。毫无疑问，数据提取的任务由数据部来承担，无论是拆分状态时还是合并一体时的数据部，数据分析师总是最繁忙的，每到月初需要支持财务经营分析的人员提取数据时，还要拉上不少工程师共同参与这场提数大战。前后台数据部拆分状态时的最后几个月里，财务经营分析的数据提取任务几乎被平均的分到了这两个部门，每到月初，提数情况便成了两个部门最直接的竞争内容，完成得不错的业务方可能会发个感谢信，当然，就算不提出批评，完成的不好的部门也会觉得很没面子。前后台数据部合并后最疯狂日子里，数据部实行“全员提数”，业务方通过公司的服务管理平台发起数据提取需求，部门内几名负责人会将需求分配给不同



的人来处理。一段时间内，数据提取的数量和难易度甚至与绩效考核挂钩，提数，成了大家工作中非常重要的一部分。

正是在这样的背景下，数据集成开发平台的 1.0 版本诞生了。这是一款查询数据并且支持周期性数据订阅的产品，同时打通了京东私有云服务 Jbox，可以供已授权人员安全、便捷的查询和提取数据，尤其对于需要定期提取大量数据做分析的人员（如财务经营分析同事）有很大帮助。功能上来讲，通过 Web 端在线数据查询和数据订阅是两大主要功能，同时，SQL 编写界面还支持元数据信息的查看，并且可以在线保存编辑中的代码，这给提数人员带来很大便利。底层接入的数据库包括当时存在的 SQL Server、MySQL 和 Hive，SQL 语法上根据不同的数据库类型选择不同的语法即可，其他执行逻辑都是一样的。产品开发主要由彦伟团队负责，从初期版本上线到后期功能升级，Tiger 也给出了很多极具参考价值的意见，不断完善的产品吸引了大量用户的注册使用。

同样采用 Extjs 的前端页面偶尔存在一些滚动条失灵的小 Bug，这给用户体验上带来一定影响。另外，虽然 Extjs 强大的表单功能成就了集成开发平台这样的富客户端应用，但是其 UI 风格的局限性也是非常明显的。随着后期产品线的丰富，新推出的产品已经弃用 Extjs，转而采用 Bootstrap 前端，于是在 2014 年 7 月，采用新的前端技术，数据集成开发平台与后期推出的数据知识管理及数据质量监控产品融合后统一在一个系统上线。

## 数据知识管理平台

数据知识管理平台产品的出现是个水到渠成的结果，在数据仓库模型规范确定之后，元数据信息也有了标准的分类体系。按照标准的分类体系将元数据信息分门别类管理起来，同时提供内容搜索、类 Wiki 的编辑维护以及咨询评论功能，数据知识管理平台就呈现在大家面前了。后期版本升级过程中又提供了维度表的维护功能，给模型开发维护的同事带来很大便利。

## 京东分析师

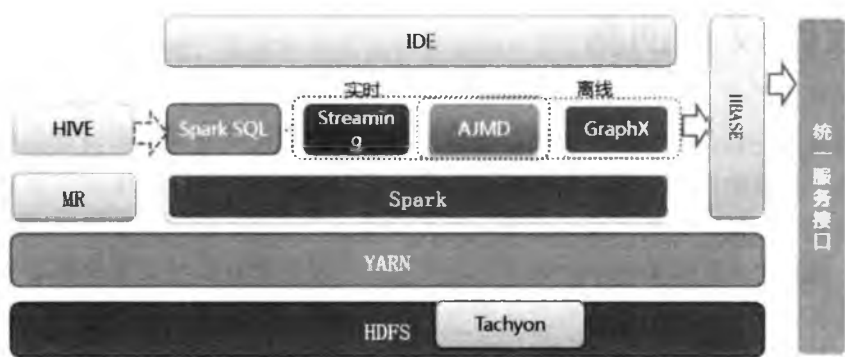
Apricot（杏子）、Blueberry（蓝莓）、Cloudberry（云莓），水果连连看？不，这是报表展现平台三个版本命名的代号，也是产品域名的首段字符串，首字母分别是 ABC 也代表了产品演进的过程。当前版本代号为 Cloudberry，产品正式名称为京东分析师，毫无疑问，我们赋予这款产品的除了基本的数据可视化能力，还有数据分析的能力。体验过 Tableau 的用户都会被其灵活的控制台和美妙绝伦的图表展现所征服。我们所做的就是在 Web 系统中尽可能的实现 Tableau 桌面系

统所能达到的效果，并且在产品服务能力上更加强调自助服务的智能软件分析平台。

技术架构上，京东分析师前端自主开发可自定义的展现布局，封装了丰富的图表展现组件，后端报表配置系统支持 MySQL、SQL Server、Oracle、API 及 Hive 等作为数据源，并支持在线接入。交互方面，报表收藏、基于图表的条件过滤、数据排序、深度钻取是其基本功能，自定义报表页面还提供邮件推送报表的功能，当某个报表数据比较重要，系统可通过邮件的形式定期发送报表供查阅。对于自己权限范围内可浏览的表，系统还可根据浏览记录将经常查看的表排在靠前的位置以提升体验。

## 数据挖掘平台

大数据的数据挖掘与传统意义上的处理方法存在很大区别，京东数据挖掘平台产品定位于构建一站式的数据挖掘算法平台，在基础的机器学习算法之上，可根据具体实际业务开发订制算法，满足算法应用场景。这一产品主要利用分布式计算，采取适用于机器学习算法的计算模型进行迭代，以解决大数据量的算法处理问题。平台封装的 Cross Validation（交叉检验）、Grid Search（网格搜索）等基本数据挖掘流程给数据挖掘人员提供简单、易用的挖掘工具。



数据挖掘平台架构

为减少数据实体化的开销，挖掘平台采用基于内存的存储引擎，集群资源调度与管理基于 Hadoop Yarn 框架，保证了集群计算性能的高可利用性和高可扩展性。平台自 2014 年年中正式推出后，已经开始为广告系统、推荐系统等提供个性化的数据挖掘算法服务。

## 数据质量监控平台

数据的及时性、准确性和完整性关系到一系列数据应用的效果，大数据平台建设之初便已着手实施数据治理的相关工作，统一数据计算口径，设置数据校验规则，以保证数据质量。数据仓库升级之后，对于数据质量的关注程度更高，于是便从产品层面进行管理。从数据生产过程来看，数据质量监控平台的基本功能包括数据生产过程中的质量检验、数据入库后的质量评估以及全部生产日志的扫描存档并生成数据质量分析报告。数据生产过程中的质量监控主要对数据生产中源表结构的变化、字段信息的一致性进行规则校验，并依据校验结果进行质量评估，对存在质量问题的数据将进行自动重跑并通知后续依赖任务。入库之后的数据将进行具体到字段粒度的数据检查，可以对枚举值、字段类型，甚至数值型字段的最大最小值及均值等进行规则校验，以确定数据是否在合理的范围内变化。

## 大数据技术创新

当前，业界对大数据处理的技术方案几乎已经形成一套通用的技术架构。然而，结合不同的行业及不同的实际业务需求，大数据平台建设所用核心技术依然需要付出一定的研发投入来实现定向的技术突破与应用。京东大数据平台搭建过程中，遵从“大处着眼，小处着手”的原则，在通用的开源技术架构基础上进行定制开发和产品化，从技术和产品两个层面实现微创新和突破性创新，打造坚实的平台基础。

技术微创新主要体现在平台的基础架构上，通过对 Hadoop 分布式技术的深入学习，在本地化部署过程中实现多项创新：

1. 定向开发，支持基于 HDFS 创建索引；
2. 定向改造，在 Map 层面合并小文件，节省资源消耗；
3. 支持动态调配资源队列；
4. 对任务运行时间和 HDFS 的读写量支持通过参数进行控制；
5. 自主开发适用于京东大数据服务平台的计算函数；

## 6. 优化并提升 Hadoop 存储性能，支持更多类型的数据存储格式：

产品层面的突破性创新主要体现在大数据管理应用上，包括：

1. 自主研发面向复杂业务模型的海量数据生产调度管理平台；
2. 自主研发打通广告点击与页面浏览的日志分析平台，同时也支持实时反作弊和用户模型研究；
3. 免推送至关系型数据库直接基于文件系统生成报表的前端展现平台；
4. 综合多种开源技术全新打造的流式计算平台；
5. 自主算法的数据挖掘平台；
6. 融合多产品功能的移动应用；

在基础架构方面，京东大数据平台是基于云计算的动态弹性平台，采用了 Hadoop 分布式计算技术，可做到快速的横向扩展。平台计算技术的实现分为离线计算模块与实时计算模块，基于海量数据查询的计算正由离线逐步过渡到即时查询，数据洞察的能力更加及时有效。

在软件架构方面，基于 Hadoop 自主开发的大数据平台，已为京东提供了高性能、高保障、高可用的通用服务，为京东数据科学家、数据分析师、数据挖掘工程师等提供了强有力的支撑，通过友好的交互设计和功能定制大幅降低了数据的消费门槛，让数据应用更简单、高效。

在业务模型方面，京东作为电商行业的领先者，拥有从流量、仓储、采购、销售、配送，售后、商品等全链路的超优质结构化电商数据。在业务模型的设计与开发方面，将传统业务主题模型进行了全方位的优化与创新，让电商主题模型更加清晰，开发更加高效。主题模型目前已经受住了复杂的京东业务，尤其是在京东业务系统频繁变更的今天，更是发挥着越来越重要的作用。

在人员配备与人才培养方面，京东复杂的业务对京东数据工程师要求更高，甚至苛刻。京东目前拥有多名数据科学家和近百名高级数据开发工程师。并通过“大数据平台认证”项目，从分布式开发技术、HSQL 优化、产品使用、业务模型 4 个方面进行专项技能培训，保证了在技术与产品迅猛改进的同时，为京东培养了更多的非数据序列的员工，通过智能数据分析平台，协助业务人员更有效地解决实际业务问题。目前大数据认证项目，已经为京东输出了数百名合格的数据工程师和数据分析师。

## 海量实时数据技术架构

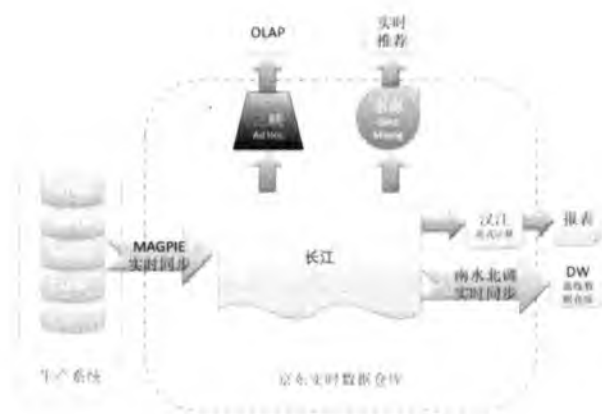
快速决策必然需要快速的数据支持，无论是京东内部的运营者还是在网站上消费的用户，实时数据太重要了。大数据平台开始实时数据技术研究应用是从2012年下半年开始的，阿侃是最早涉足这一领域的同事之一。近两年的时间里，相关技术方案先后在618、双11等重大促销活动中投入使用，进行PV、UV、订单量的实时查询。2013年年5月，京东用户在页面上进行产品搜索时，会发现搜索结果的前面显示了“已购买”、“已浏览”等这样的标签，这种个性化的搜索结果展现也是实时大数据计算应用的一个方向。



实时数据计算网页应用

京东大数据平台实时计算分为四个阶段，分别采用了不同的技术。

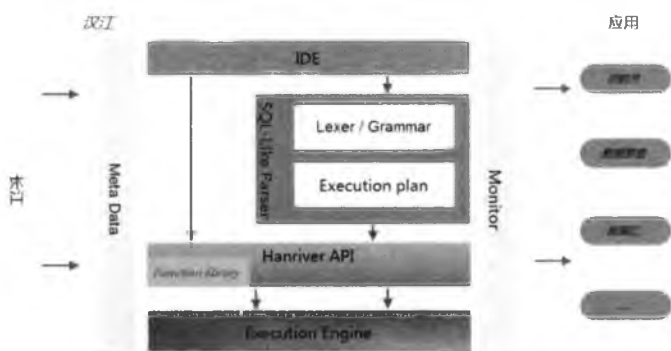
1. **对于实时数据采集阶段。**成立了一个项目“长江”，将异构数据源实时同步到实时数据仓库里。“南水北调”项目同时将数据同步到离线数据仓库里。
2. **流式数据计算阶段。**承接“长江”项目，成立了“汉江”项目。“汉江”项目将 storm 做了更好的封装，可以更方便的调用实时计算 API。
3. **实时查询服务。**经过前面两步，数据已经实时计算并存储了。“三峡”项目应运而生。“三峡”使用 Spark, PrestoDB 来解决海量数据下的即席查询。上亿条数据，简单的查询，可以在十几秒内返回结果。
4. **实时数据挖掘。**“东海”项目结合“长江”项目的实时数据，将数据挖掘实时化，更好地服务线上的推荐、搜索。



实时计算架构

在京东的业务场景里面，有大量流数据的实时计算需求场景，大数据平台提供了一套通用的流数据实时计算平台，内部称之为“汉江”。汉江平台实现了计算过程的低延迟、高吞吐，计算结果的可复用和计算任务开发的简单化的目标，采用复合事件模型构建分布式流数据计算框架（底层基于 Storm），功能易扩展、部分容错、数据和状态可监控。汉江具有处理实时数据流（长江产品实时从生产环境中多个异构数据源同步的数据）的能力，同时将实时计算结果以服务的方式供业务系统调用，支撑业务团队的个性化需求。目前，汉江平台为京东数据罗盘、京东数据领航员、京东数据汇等产品提供实时计算数据。

结合京东的业务特点，汉江平台对外提供两种使用模式：SQL-Like 和 API 编程，并内置了大量可复用的函数（如：Distinct、Sum、Avg、Count、Join 等），结合业务提供了一些业务化的函数（如：PV、UV）。



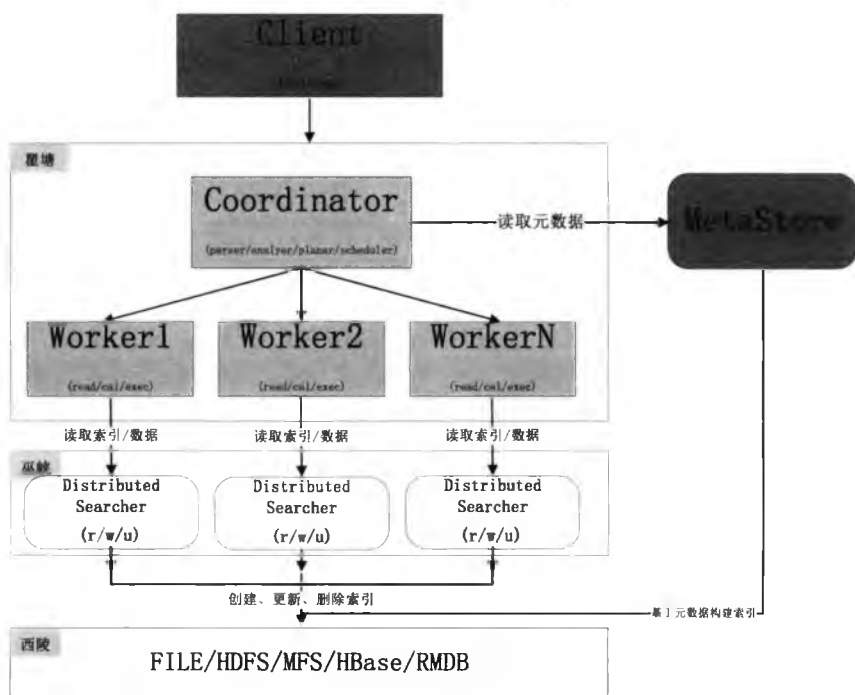
流式计算架构

## 海量数据实时查询技术框架

京东的海量数据实时查询产品取名为“三峡”。“三峡”是一个基于分布式的支持以 SQL 的形式提供数据查询、针对亿级数据支持秒级响应,资源占用率较低。

技术架构整体分查询层、计算层、存储层。先由查询层接收 SQL 语句并将其根据元数据信息解析成后端查询语言,再将查询语言提交给计算层,由计算层将查询请求分解为多个 Job 并分发到计算集群当中的所有数据计算节点,由数据计算节点执行 Job,期间会与存储层发生数据交互,最后将计算后的结果返回,由计算层将多个数据计算节点计算的结果合并后返回给查询层,完成整个海量数据实时查询过程。

整个技术框架中囊括的技术点包括:Socket 通信、SQL 语法解析、Cache 集群、并行计算集群、索引、列存储、元数据。



海量数据实时查询架构截图

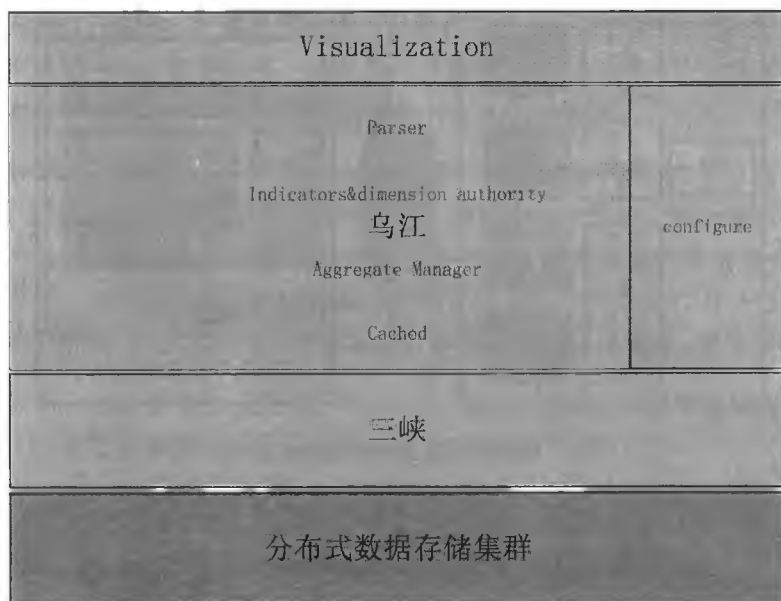
1. **Client:** 是一个基于命令行的客户端,用于接收用户输入、执行 SQL,查看返回结果,了解执行状况的窗口;
2. **瞿塘:** 负责解析 SQL、生产查询计划、执行数据计算,计算结果合并;

3. 巫峡：负责索引构建、查询、删除和优化；
4. 西陵：负责数据存储、备份；
5. **MetaStore**：负责元数据管理。

## 基于分布式架构的联机分析处理（OLAP）技术

大数据量的快速、复杂查询处理需要 OLAP 技术来支持，但面对京东庞大的数据量，传统的 OLAP 技术显然没有用武之地。我们曾经尝试使用 Hadoop 平台，通过使用 Hive 模仿出 SQL 来做 OLAP 的数据分析，在 Hive 中虽然将 HiveQL 翻译 MapReduce 的时候进行了优化，但依然效率低下。多维分析时依然要做事实表和维度表的关联，维度一多性能必然大幅下降。除了查询性能方面的问题，目前 OLAP 存在的最大问题还有：业务灵活多变，必然导致业务模型随之经常发生变化，而业务维度和度量一旦发生变化，技术人员需要把整个 Cube（多维立方体）重新定义并重新生成，业务人员只能在此 Cube 上进行多维分析，这样就限制了业务人员快速改变问题分析的角度，从而使所谓的 BI 系统成为死板的日常报表系统。

基于探索和尝试我们研发了自己的基于分布式架构的联机分析处理系统，如下图：



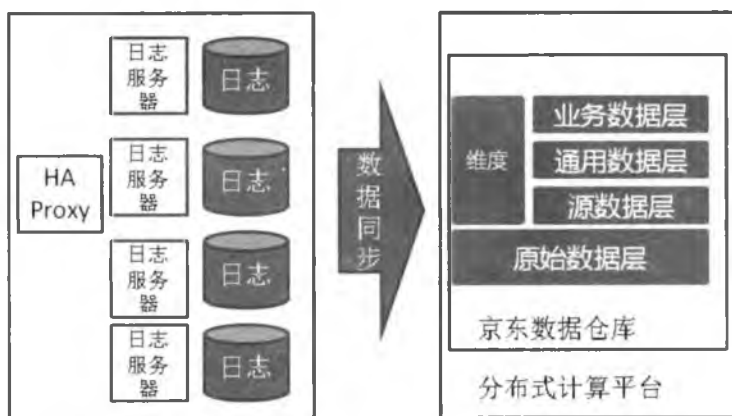
基于分布式的 OLAP 架构



Visualization 是一套适合京东业务使用方式的前端可视化框架。代号为乌江的系统是指标维度映射管理、权限管理、CUBE 创建以及数据汇总和缓存的服务系统。其中, Parser 是逻辑语句的解析模块, Configure 负责指标维度的管理、权限管理、逻辑 CUBE 的管理, 用户可以灵活的调整, 使适应业务灵活多变性。Aggregate Manager 是聚合逻辑处理模块。高性能的分布式数据查询引擎“三峡”前文已经有所介绍。底层分布式数据存储集群, 是具有可以任意修改和扩展指标和维度的数据存储格式的数据集群。

## 海量日志数据分析处理技术

作为一个互联网公司, 京东有大量的业务是在线的, 用户行为日志的数据可以有效地帮助京东改进系统, 提高用户体验。然而海量数据的处理和分析是一个挑战。针对这个情况, 我们采用了页面的 js+HA Proxy+专用日志服务器的方案解决日志收集的需求。js 定义数据, 可以和用户行为一一对应上, 并且可以有更多的定制化信息可以记录。HA Proxy 保证了在大数据量的情况下, 采用负载均衡的分发机制, 将日志的写请求对应到不同的日志服务器。在数据收集层面, 采用了专用的日志服务器, 将日志写请求转换为标准的文件写请求。最大化地利用了服务器硬件效率。



海量日志数据处理架构截图

在上述的数据收集完成之后, 我们会将这些日志文件同步到我们的分布式数据处理平台。在分布式数据处理平台上, 我们对数据进行清洗转换, 加载其他有效信息, 如订单、用户等, 将日志加载成标准的范式定义的数据, 提供给下游业务自定义使用。

## 京东大数据创新应用篇

### 基于大数据的精准营销

Amy 是京东的一个新会员，她最近想当一个家庭主妇，因为想自己开始学一些厨艺，于是到京东去买些厨具，结果发现她选中的那款商品没有货，然后她看到京东有“到货提醒”功能，于是他选择了该功能，填上了自己常用的邮箱地址，然后确认，相当于登记好了。过了几天这个商品有货了。Amy 就收到了一封邮件说，亲爱的用户，你上次想买的东西有货了，你要不要买，然后还在这封邮件里给她推荐了几个相关的商品。可能由于很多原因，Amy 改变主意了，感觉自己选的没有推荐的好，于是她购买了邮件中推荐的商品，通过邮件 Amy 完成了她在京东的第一次购物。

过了一段时间，Amy 又迷恋上了摄影，于是想在京东买一款单反相机，她搜索浏览了很长时间，但对于一个摄影菜鸟来说，她一直不知道该如何选择。没想到有一天她打开邮箱，发现里面躺着一封邮件“京东告诉您如何挑选单反相机”，呀！这不正是 Amy 需要的吗！她立马打开邮件，通过邮件到达专题页面，参照里面的内容，果然找到了自己满意的相机并果断下单购买。

Amy 的爸爸快要过生日了，她打算送爸爸一部三星手机，在京东有一部她感觉不错，就是价格有些贵，Amy 有些犹豫，先放到购物车吧，再看看有没有其他的便宜一些的，但是当天也没有找到更合适的，正好有其他的事情就去忙去了。结果3天后，她收到一封“您购物车里的商品降价啦”的邮件打开一看，就是她想买的那个手机，降价500元，降价后的价格他觉得可以接受了，就果断购买了。

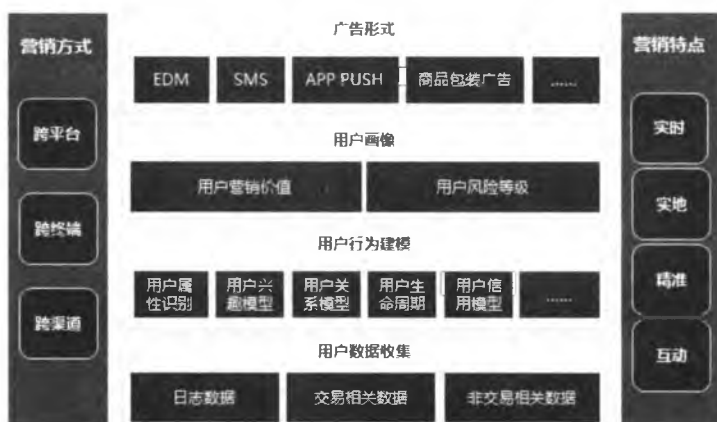
就这样，Amy 喜欢上了京东的邮件，因为京东的邮件总能给她惊喜，能帮助她购物，好像能读懂她的心思，这是在其他家网站没有的。

像 Amy 这样感受到京东邮件魅力的会员不在少数，那么京东的邮件系统是怎么做到的呢？

首先，我们知道好的邮件营销就是要完美解决一个3W的问题：即在什么时间（When）把什么内容（What）发给什么人（Who）。如果要解决这个问题，就要很清楚地了解用户的情况，用户的个人喜好，他需要什么，这就需要大数据挖

掘技术的支持，需要基于用户在京东的一切行为（行为背后是一系列的数据），包括搜索、浏览、点击、咨询、加关注、放购物车、下单、地址等等一系列数据，在这些数据的基础上进行建模，然后我们得出每个用户的情况，例如：性别，年龄，婚否，是否有孩子，孩子的性别，是否有房子，是否有车，喜欢什么品牌，等等，当我们了解了这些信息，就比较容易定位到每个用户的喜好。然后我们再抽象出各种场景，基于每个场景制定不同的邮件策略，例如：加购物车却没有购买是一个场景，例如浏览了什么什么东西没有购买也是一个场景，那基于这些场景，我们设置不同的邮件内容，在合适的时间，例如加购物车后这个商品发生了降价行为的时候，发送给这个用户。

基于大数据京东搭建了以下结构的精准营销架构：



精准营销架构

架构底层是明细数据，包括用户产生的各种日志数据、用户交易数据和其他相关数据、在用户数据的基础上、我们进行了用户行为的建模，包括用户属性的识别，用户兴趣，用户关系模型、用户生命周期、用户信用模型等等；在用户建模之上，我们抽象出用户画像，作为底层数据供应给各营销系统。

所以这个架构解决的不仅是京东邮件精准营销的问题，一起解决了所有主动推送的，例如：短信、App PUSH、站内信等的精准营销问题。

在京东的邮件系统还没有引入大数据之前，只是每周发两次全站用户的邮件，或者有一些简单的基于用户级别的划分，当时负责邮件营销的小新很是苦恼，因为他有很多邮件创意无法实现。有了大数据的参与，我们把大数据建模出的用户画像抽象成筛选条件放到邮件系统，这样，任何邮件运营人员都可以很方便地筛

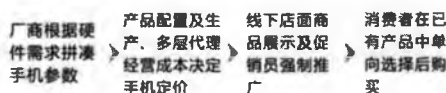
选出精准的目标用户，邮件内容的设置上也更加多样化，更重要的是用户体验得到了极大提升。在基于大数据做了很多场景的自动触发邮件策略之后，京东的邮件也开始变得智能起来。

## 电商大数据 C2B 定制

2014 年 3 月，国家主席习近平和夫人彭丽媛在德国观看中德少年足球友谊赛的时候，彭丽媛使用了一款国产手机现场拍照。国母用的手机，想不火都难！于是一段时间内掀起了一股“国母手机”热潮。究竟什么手机能这么火，都惊动国母了呢？这还得从 2013 年京东提出的 JDPHONE 计划说起。

JDPHONE 计划是京东在 2013 年提出的以用户需求为中心的创新型计划。通过京东海量用户数据进行深入分析，挖掘用户真实需求趋势，联合品牌厂商，整合产业资源，共同打造满足用户需求、超出用户期望高性价比的产品。这种从消费者需求出发，驱动制造业进行产品设计与生产的模式正是目前流行的反向电子商务模式（C2B）

传统手机生产模式



JDPHONE 计划与传统手机生产模式对比图

京东的 C2B 大数据定制分析系统：“慧眼”

1. **整体思路**: 市场是否饱和, 机会在哪? 用户的需求及趋势是什么? 产品如何生产能满足用户期望? 如果投入生产预计销量如何? 产品后期营销效果如何? 是否达到预期目标?
2. **市场分析**: 分析市场规模、饱和度、潜力, 分析各品牌市场占有率, 让生产商了解市场现状与未来发展趋势。
3. **消费者分析**: 分析消费者的结构、消费习惯、需求及满意度评价, 让生产厂商了解客户的各方面情况, 从而更好地进行产品设计。
4. **商品分析**: 把商品进行属性标签化, 从属性的维度对标品、非标品进行细分, 建立分析不同人群需求下的规格组合, 帮助生产厂商进行深度的产品定制。



JDPhone 计划流程

拿手机为例, 近半年来在京东购买过 2 次以上手机的用户, 其中 34% 的人在每两次手机更换时都会选择更大屏幕, 但是 5.5 寸手机屏幕是他们接受的极限, 可能是因为中国人的手较小, 如果屏幕再大的话一个手估计掌握不了, 因此系统会建议选择 4.8~5.5 寸屏幕作为最优尺寸。500~999 价位的手机销量最多的地区是广州, 很多是来自广州地区制造业加工厂的外来务工人员, 他们的消费能力有限, 平均手机消费较低, 并且因为他们大多数都来自外乡, 经常会给家乡的亲戚

打长途电话，因此他们很迫切地希望能用低廉的长途通信费用业务。因此系统会建议面向广州地区的手机营销策略为：主打中低端价位手机，性价比高，结合京东虚拟运营商业务，降低长途及漫游费用。

京东大数据平台数据创新部从价格分布、关键属性、流量、成交量、消费者评价等维度建模，挖掘出功能卖点、主流价格段分布、消费者需求、增值卖点来指导厂家的研发、设计、生产。目前 JDphone 计划已经成功地推出了：努比亚大牛、小牛 2、华为荣耀 3C、诺基亚 X、IUNI U2、努比亚牛魔王等多款产品，每款产品的月销量都过万。

传统的 C2B 模式，互动、调研、预售、团购、定制、选配都是主动行为，用户都需要主动参与其中。而在京东 C2B 大数据分析过程中，我们只需要针对不同场景建立不同的分析模型，然后准备好相应的数据源，系统便能自动计算出我们想要的答案，用户是被动参与整个过程的。我们不但以用户需求为中心，并且让用户参与的成本被降到最低，这种模式将会是未来的发展趋势。

## 用户画像

为什么要做用户画像？一方面是海量信息的汇集，京东是一家大型全品类综合电商，海量商品和消费者产生了从网站前端浏览、搜索、评价、交易到网站后端支付、收货、客服等多维度全覆盖的数据体系；另一方面日益复杂的业务场景和逻辑使得信息的处理挖掘日益重要。也就是说，京东已经形成一个储量丰富、品位上乘且增量巨大的数据金矿，但是在相当长一段时间，很多业务人员经常面对宝山空回的局面，比如我们的数据分析师和工程师被业务人员反复追问，为什么我的促销活动做了这么久，力度也挺大，就是没有带来预期用户的增长呢？从用户画像分析来看，很可能是在错误的时间错误的地点对错误的人做了错误的促销活动。

用户画像就是在解决把数据转化为商业价值的问题，就是从海量数据中来挖金炼银。这些以 TB 计的高质量多维数据记录着用户长期大量的网络行为，用户画像据此来还原用户的属性特征、社会背景、兴趣喜好，甚至还能揭示内心需求、性格特点、社交人群等潜在属性。了解了用户各种消费行为和需求，精准刻画人群特征，并针对特定业务场景进行用户特征不同维度的聚合，就可以把原本冷冰冰的数据复原成栩栩如生的用户形象，从而指导和驱动业务场景和运营，发现和把握蕴藏在细分海量用户中的巨大商机。

从逻辑上说，是从具体的业务场景出发，结合数据表现，归纳出基准的规则

或方法，然后通过反复迭代的学习过程，生成符合既定约束条件的最优化方案，然后把此方案泛化推广到类似的场景中去。很多时候用户画像都是从一个具体品类的业务场景或需求出发，有一些业务人员运营经验很丰富，结合对他们经验和需求的访谈，工程师会把业务语言抽象出来，结合数据语言转化成通用的技术语言，然后用他们神奇的大脑和给力的大数据平台生产出符合需求预期的结果，经过业务人员反复验证有效后，这个画像就宣告成功，然后，工程师会再次驱动其神奇的大脑，将此画像推广到京东全站应用中去，这种从群众中来到群众中去的方法因其敏捷高效快速迭代的优点，产生了一大批性能优良的产品。

当然，对于一些用户画像基本属性来说，由于其对所有品类或场景的通用性较强，跳过单品类测试，直接针对全站用户建模的做法，效果也非常好。

用户画像应用服务支持京东集团全业务需求，其下游面向的人群工程师需求各不相同，从技术方案到使用方法也千差万别，因此有必要采取体系化多层次服务平台进行支持。对于公司内部，针对研发、采销、市场、客服、物流等各体系不同需求分别采取统一数据仓库、数据接口服务、产品化平台多种服务方式提供支持，针对各业务线需求场景不同，人员经验也不尽相同，用户画像的平台化给内部使用人员打造了切合自身业务场景和使用经验的操作：对经验丰富的使用者提供更深入的综合参考，并允许其自主订制或二次开发。给经验较浅的用户提供数据之外还培养其分析意识；对小白用户则可建立数据化分析运营的意识与习惯；对外部用户的支持力度也在逐步放开、加大，比如 POP 商家，可以满足商家针对自身店铺的个性化订制需求，并结合各种营销方式提供一站式服务解决方案。

在京东用户行为日志中，每天记录着数以亿计的用户来访及海量行为。我们通过对用户行为数据进行分析和挖掘，发掘用户的偏好，逐步勾勒出用户的画像。用户画像通常通过业务经验和建立模型相结合的方法来实现，但有主次之分，有些画像更偏重于业务经验的判断，有些画像则更偏重于建立模型。

业务经验结合大数据分析勾画出的人群，其用户画像由于跟业务紧密相关，更多的是通过业务人员所提供的经验来描述用户偏好。举个例子，比如：根据业务人员的经验，基于客户对金额、利润、信用等方面的贡献，建立多层综合指标体系，从而对用户的价值进行分级，生成用户价值的画像。一方面我们的产品经理可以根据用户价值的不同采取针对性的营销策略，另一方面通过分析我们的不同价值等级用户的占比，从而思考如何将低价值的用户发展成高价值的用户。

再比如，通过用户在下单前的浏览情况，业务人员可以区分用户的购物性格。有些用户总是在短时间内比较了少量的商品就下单，那么他的购物性格便是冲动

型；有些用户总是在反复不停的比较少量同类商品最后才下单，那么他的购物性格便是理性型；有些用户总是长时间大量的浏览了很多商品最后才下单，那么他的购物性格便是犹豫型。对于不同购物性格的用户，我们可以推荐不同类型的商品。针对冲动型用户，我们直接推荐给他/她最畅销的同类商品，而理性型用户我们推荐给他/她口碑最好的商品，并且针对每一个用户，我们根据其购物性格定制了个性化的营销手段。

以建立模型为主勾画的人群，我们不能认为买过母婴类用品的用户家里就一定有小孩，因为这次购买很有可能是替别人代买或者送礼物。所以我们要判断这个用户所购买的母婴类商品是否是给自己买。根据用户下单前浏览情况、收货地址、对商品的评价等多种信息建立模型，最终判断出用户家庭是否有小孩。再根据购买的商品标签，比如奶粉的段数、童书适应年龄段等信息，建立孩子成长模型，在孩子所处不同的阶段进行精准营销。

京东拥有最全的品类，各品类间用户转化成为我们业务的一个重点。挖掘一个品类的潜在用户，首先要找出此品类已有的用户，然后通过这些用户的行为、偏好、画像等信息对用户细分，挖掘其独有的特征，最后通过这些特征建立模型定位出该品类的潜在用户。

这一阶段主要是为了验证我们为用户描绘的画像是否准确。比如一个用户的画像是：性别男、年龄在 36~45 岁之间、家里有小孩、未婚、有车一族、购买等级高。我们可以很快发现家里有小孩且未婚这一矛盾的结果。首先，我们可以判断这个用户的画像肯定有问题。接下来我们看这个用户的画像，似乎只有未婚这一条与其他画像格格不入。通过模型之间的验证，我们发现一些错误案例并分析原因，进而改进我们的模型。

用户画像提供统一数据服务接口供网站其他产品调用，提高与用户间的沟通效率、提升用户体验。比如提供给推荐搜索调用，针对不同用户属性特征、性格特点或行为习惯在他搜索或点击时展示符合该用户特点和偏好的商品，给用户以友好舒适的购买体验，能很大程度上提高用户的购买转化率甚至重复购买，对提高用户忠诚度和用户黏性有很大帮助；再比如数据接口提供给网站智能机器人 JIMI，可以基于用户画像的用户量身定做咨询应答策略，如快速理解用户意图、针对性商品评测或商品推荐、个性化关怀等，大幅提升 JIMI 智能水平和服务力度，赢得用户欢迎和肯定。

今年 618 前夕京东产品的数据接口服务，将用户画像模型充分应用到产品当中，根据族群的差异化特征，帮助业务部门找到营销机会、运营方向，全面提高



了产品的核心影响力，增强了产品的用户体验。包括年龄、性格、购物偏好、购买力等用户特征的应用模型，诠释勾勒出用户在京东上的体貌特征，赋予一定的潮流“范儿”，贴近用户。

明嘉靖朝著名内阁首辅徐阶主政后，在其工位上有个著名条幅，其著名程度堪与东哥在苏州街墙上题的“战斗！战斗！只争第一”相媲美，他写的是“以威福还主上，以政务还诸司，以用舍刑赏还诸公论”，我的理解，就是各司其职，各展其长，用户画像的服务也是针对各服务对象进行区别对待，方便用户使用，具体实现方法是：JDW+CUBE+BMP。

首先，对用户画像的结果进行标准化加工，同步至企业统一的 JDW 平台，解决数据孤岛，方便研发底层调用；同时，按主题建立多维分析的数据 CUBE，直接面向分析师和工程师；另外，还进一步打通上下游关联数据和产品，尤其是大营销平台（BMP），这个主要针对产品经理和一线采销人员，他们可以在 CUBE 筛选出预定人群后直接调用营销平台进行发券、EDM 等操作，减少了诸多中间环节，实现高效运营和精准营销，大大提升人效。其中，多维分析 CUBE 是用户画像产品化的出色应用之一，其用户画像的诸多维度和订单、商品、流量等指标的组合可以快速实现智能分析，并可根据数据对比分析提供专业有效的建议，将数据转化成知识和决策供大家使用。

京东数聚汇通过深度分析年度网购用户的行为，挖掘网络购物趣味数据，结合用户画像，从用户的购物行为入手，结合年度流行热点，分析不同地域网购人群的购物习惯和喜好，为网民展现一场京东大数据的饕餮盛宴，同时给商家和消费者提供了经营和购物参考。2012 年京东数聚汇，通过简洁的图片，数据可视化的展现形式，深度分析了 2012 年度网购用户的购物行为。

在电商逐渐占据人们生活的今天，这些电商自己提供的“电商数据”不仅是一个很好的年终盘点，同时也对未来的发展提供启迪。而且数据趣味性也是大数据展示的重要部分。例如：平均每 100 个程序员中就有 52 个人会购买《给心灵洗个澡》这本书。看来程序员很喜欢在辛勤工作之余给自己的心灵洗个澡，以此来缓解内心的焦虑。

2013 年数聚汇以“大数据诠释流行”为主题开启。《魔兽世界》玩家是成为奶爸的潜力股？土豪识别不能光看 5s？中国大妈背后其实是中国大爷？四川男女最招羡慕嫉妒恨？足球场内不行，场外也“性”趣不高……如果说这些是刚刚经历的 2013 年，你信吗？伴随这一个个热门的话题，你可以看到数据背后你不知道的那些事。

## 店铺评级

纵观古今，级别的概念充斥着生活的每个角度。从个人的银行信用到国家的安全等级，从显示器的清晰度到每个人的工资水平。级别是迎合人们对“标准”需求的一种产物，更是被众人当作衡量标准的尺子大用特用。在电商领域，随着业务范围和平台模式日益扩大，店铺量日以万计迅速增长。面对这么一个庞大的群体，我们是否也需要设定某个标准来衡量店铺的水平高低呢？这样的一把尺子应该怎么设计呢？

信息对称和公平交易是关系密切的两个概念，信息不对称则普遍存在于用户和店铺之间，早已成为各种不平等交易的诱因。当用户缺乏店铺服务水平的相关信息时，就会影响用户对店铺的信任程度，进而影响用户做出“进与不进”和“买与不买”的决定。

在店铺另一端的运营人员对于信息的需求则同样刚性。所谓的“酒香也怕巷子深”，挂上一块“物美价廉，服务优质”的招牌是他们对信息共享的迫切需要。日常运营中哪里做得好、哪里还需要优化、用户对各环节的认可程度如何，自家在行业中处于什么位置，这些也是运营人员亟待了解的信息，对于运营有着重要的指导意义。

处理信息的方式多种多样，量化信息是其中最直观也是容易被理解的信息存储和展示方式，也是运营管理中可以直接用于评估考核的一种方式，能够快捷地为信息使用者提供含义明确、尺度清晰的内容。信息对称和共享，庞大的群体需要更规范更科学的管理，这些使得找到一种衡量店铺众生相、承载店铺优劣信息的方式成为必然。店铺级别体系正是基于此而产生的使用统一标准衡量不同店铺水平、共享店铺多方面量化信息的一种载体。

搭建评级体系过程中需要考虑诸多重要因素。

**第一个因素是评估对象的准确性。**体系以体现店铺的“综合”水平作为最终目标，但“综合”是一个宽泛抽象的概念。幸运地是它能够被具体行为所表现，可以通过行为来具体化描述。用户和店铺之间发生着各种行为，如购物、网页点击、售后服务等等，这些都影响着店铺和用户之间的关系、调整着用户对店铺的观感、反映着店铺的管理效果的量化的重要影响因子。

如何从各种行为数据出发，准确定位评估对象属性特征，抽象和量化关键性因子和具体指标，是体系建立中的重要工作。采取 AHP 和因素分析法对用户购物等行为进行量化拟合后发现，商品、服务和时效是影响用户购买和评价的最主

要环节，其中商品质量、服务态度和物流速度又是各环节的主导因素。通过一系列的分析模拟，得到了体系将要评估的明确对象和各个细分因素。

第二个因素是可解读性。评级体系是以数据为基础、运用抽象思维建立起的面对真实业务群体的可视体系，容易解读是体系得到广泛认可和贯彻推广的必然要求，体系的不可理解将是体系的灾难性缺陷。评级体系对象、体系结构和数据基础与体系间的关系，都需要实际的业务或行为作为其理论基础。

体系中的“量化”方式是体系解读中较困难的部分，它选择尤为重要。当衡量店铺的某项水平时，可以采取水平的原始值标识、可以将店铺水平指标排序后的 Rank 展示给用户，或者建立水平的指数化指标。考虑到某些方式存在对数据标准化和函数处理的操作过程易造成解读困难，最终采取了易解读的方式。

第三个因素是体系结构的合理性。评级体系结构的合理性会影响到体系运转过程中的稳定。评分体系可以采取函数结构、多层非交叉结构和多层交叉等结构，不同结构各有特征和优劣。在权衡不同结构的性能后，可以为评级系统设计更恰当的体系结构。结构布局确定后，需要进一步确定结构之间的关系关联方式。在数据模型的基础上，层级之间通过模型量化计算，得到形如  $y = f(x)$  的函数式结构，作为连接各个结构模块的网络脉络。

评级体系的价值在于体系用户在使用过程中得到了帮助。

店铺评级体系上线后，前台页面及后台系统中展示了店铺评级的体系内容和各项结果，用户和运营人员能够方便快捷地找到其需要的内容，双方信息形成共享。用户清晰了解到他面前的店铺在服务、商品和时效各方面的表现和水平高低，丰富明确的量化信息为他进店和购物的决策提供了高效客观的数据支持。评级结果是运营人员提交的一张成绩单。成绩单中详细记录了店铺的优良可差，帮助运营人员找出自身的优势和不足，促进其继续努力学习。这样不仅可以改善运营死角，又能通过更好的评级成绩吸引了更多用户、提升运营效果，形成良性循环。

基于大数据研究、通过抽象量化分析建模并最终应用于网站运营的店铺评级体系，已经成为用户购物和店铺管理的重要标尺。在要求信息更加透明和管理更为客观严谨的今天，店铺评级体系得到了越来越多的应用，它将帮助网络购物变得更公平更便利。（作者/王威、陈玉兰、高慧、刘朋飞、张立军）

# 18 人工智能——智慧的 采销

## 人工智能的魅力

“可以出去旅游咯！”今天起床后，小天就一直非常兴奋，因为爸爸决定送她一台数码相机作为考上大学的奖励。小天迫不及待地进入 JD.com，迅速挑中了一台数码相机，下单的时候还不到 11 点。下午 16 点 30 分，配送员就将小天的礼物交到了她的手中。是什么力量使得京东的物流如此给力？是强大的仓储，是敏捷的物流，也是勤劳的配送员。在这些背后，还有一只看不见的大手在使这一切高效而准确，那就是京东智慧的供应链。

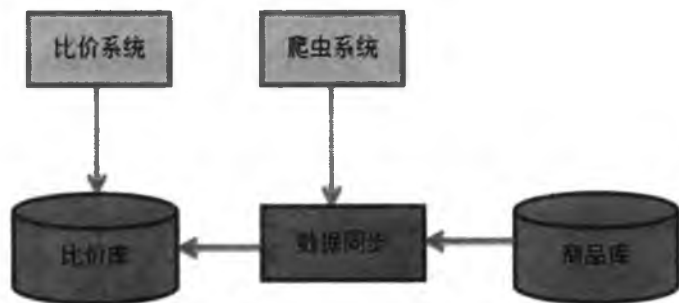
当用户在京东购买了一件商品后，用户订单会下传到库房完成生产，然后打包发往配送站，最后由配送员送货。而库房中存放的商品则是由采销人员提前向供应商采购或从其他仓库调拨好的。若库房中商品存货不足，是无法满足京东物流需要的。截至 2014 年 6 月，京东的自营商品分散在全国各大城市的上百个库房中，想要管好这么多商品的库存非常地不容易，几乎是一项人力无法完成的任务，这时，我们看到了大数据驱动供应链的魅力。

## 智慧选品与智慧定价

外行看热闹，内行看门道。今天我们来详细地跟你说说，京东是如何通过智慧的供应链，知道消费者喜欢什么商品，商品价格在什么价位最合理的。说到这里，你肯定首先会想到的就是用比价系统做到实时比价就可以了。其实不然，京东不但可以实时比价，还可以基于友商商品和京东商品相关历史数据，做大数据挖掘，提供行业竞争数据分析、热销品选品建议、智能定价等。当然，能实现这一点并不是易事，而是团队经过多年的经验积累，逐渐打造出来的。

### 2010 年

2010 年的某一天，京东的高层领导做出重要决策：上线比价系统提升用户体验。随后，便成立了比价团队。那个时候京东还在使用 .Net 语言开发，数据库大量使用 SQL Server。团队基于 .Net+SQL Server 开发了爬虫系统抓取友商商品，同时开发了 Worker 同步，定时同步京东商品信息，自动匹配 Worker 完成京东商品和友商商品的自动匹配工作。有了友商商品、京东商品、匹配等数据，比价系统上线了，但暂时只支持自营图书的实时比价和自动跟价。



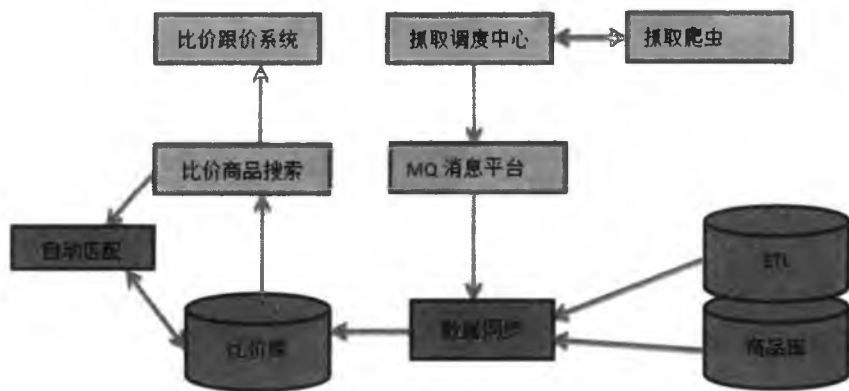
比价系统最初的架构图

### 2011 年和 2012 年

2011 年年初到 2012 年年底，京东开始全面转型 Java 平台，数据库也在向 MySQL 迁移。在这个背景下，比价团队响应公司战略，用 Java 和 MySQL 实现比价系统。同时，自营图书比价受到了图书采销的好评，其他部门采销也强烈要

求使用比价系统。比价团队为此专门立项，进行了 Java+MySQL 改造，引入了 Apache Solr 搜索技术并集成了 IK 分词器，扩展了电商行业专用词库，并针对比价的特点，自定义了 Solr 的 QueryParser，使其不但可以支撑比价中的自动匹配，同时还可以支持所有自营商品的上百万自营 SKU 的实时比价搜索、近千万 POP 商品的同店铺比价搜搜和商品不同维度的实时搜索。

为了支持所有自营商品的自动跟价，比价团队开始进行自动跟价的平台化改造。把图书跟价从比价系统中迁移出来，优化了跟价规则，基于 Web 控制台的配置和 MQ 消息解耦，实现了友商商品价格变化后京东商品实时跟价。平台化改造完成后，系统内 Worker 跟价计算、调价 Web 控制台等支持水平可扩展。



基于 Solr 与 MQ 的比价系统架构图

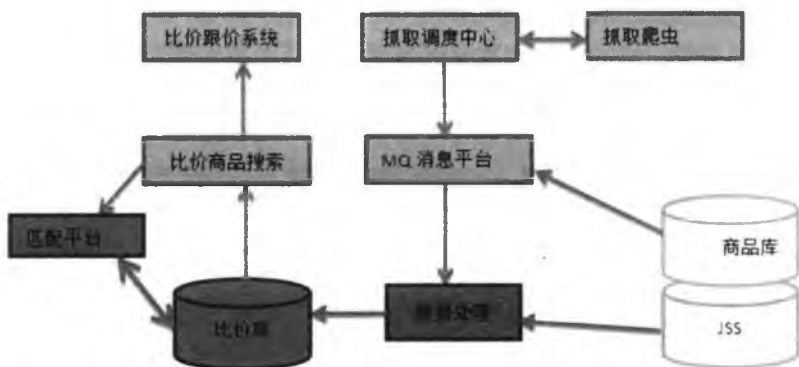
## 2013 年

2013 年年初，比价团队开始着手解决自营非图书商品的自动匹配准确率问题，一直以来，我们发现利用 Solr 内部自带的相似性计算及排序做自营商品的自动匹配，召回率虽然很高，但准确性却得不到保证，尤其是日用百货类商品。这些商品也无统一的匹配规则，不像图书类商品可以根据 ISBN+定价规则精确进行自动匹配。

自营非图书商品自动匹配有所突破的关键在哪里呢？各品类商品差异较大，自动匹配根据不同品类有不同的规则。因此，匹配的关键是友商商品与京东商品分类的统一、商品属性的归一化及文本萃取。

比价团队专门对此进行立项，把商品匹配职责从比价系统分离出来，进行平台化改造。一方面进行服务化治理，解耦合；另一方面，开发了自动匹配效果可视化工作台，算法的每次迭代效果都可借助可视化工具观测的匹配指标看出。技

术上，和数据部刘思喆团队合作，引入 SVM 机器学习算法，做商品自动分类的归一；升级了 Solr，进行分布式部署改造；同时自主开发隐马尔科夫分词器，通过词性标注、词序训练学习，提升属性萃取的准确率。以上技术结合匹配运营工作台，使匹配召回率提升到 80% 以上，准确率提升到 98% 以上，做到高 Band 重点商品全覆盖。2013 年年底，本次改造彻底上线完毕，并成功运营投产。至此，比价系统的又一重要里程碑诞生。



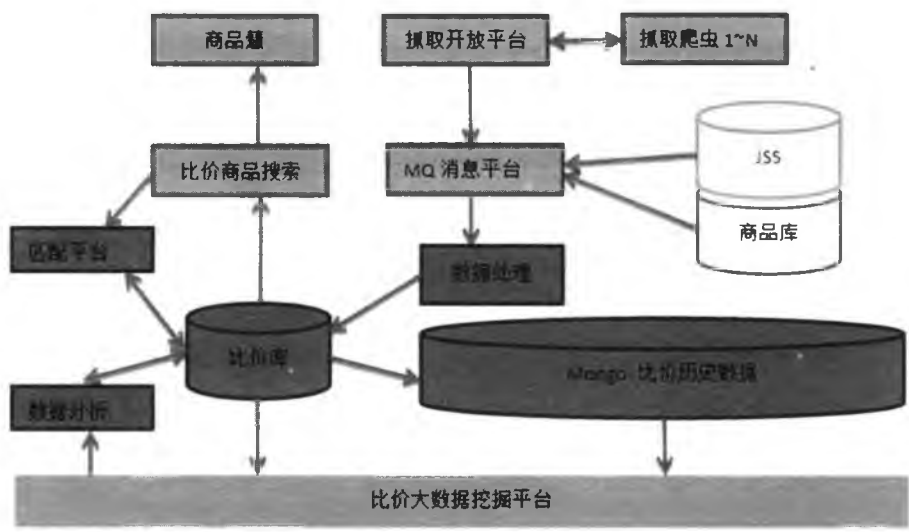
平台化改造后比价系统架构图

## 2014 年

2014 年年初，大数据驱动供应链智能化程度不断提升，比价团队也迎来了新的挑战 and 机遇。同年 6 月抓取可扩展开放平台上线，技术上基于 Mongo NOSQL 数据库的 Sharding+Replica Set，支撑日均数千万商品的调度、抓取。

通过本次架构升级，可以实现基于爬虫节点抓取能力进行灵活弹性的调度、分发抓取任务；抓取节点可随意扩展，支持部署到任意机房、地点；所有模块可水平扩展，极大增强了爬虫系统的可扩展性。一句话总结，需要抓什么、需要多长时间抓取完毕等对于抓取可扩展开放平台来说都不存在技术问题。

2014 年 7 月，比价系统三年磨一剑，京东及友商商品数据、价格历史、销量数据、商品属性数据等应有尽有，基于 Mongo+Hadoop+Hive+Spark，引入 Scala 开发语言，实时深度机器学习、行业竞争情报分析、智慧选品、智慧定价，商品智慧产品重磅推出，比价团队为公司的大数据战略添砖加瓦，为智慧供应链出了一份力！



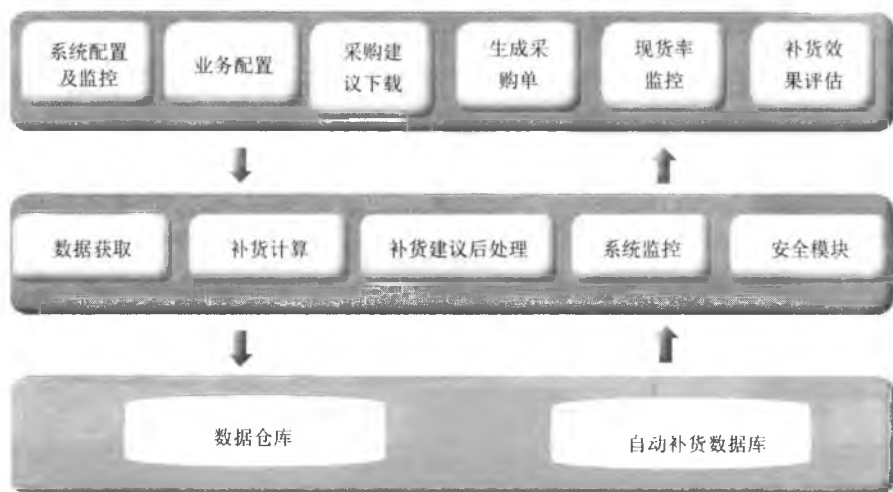
大数据环境下比价系统架构图

## 供应链销量预测

“市场上只有供应链而没有企业，真正的竞争不是企业与企业之间的竞争，而是供应链与供应链之间的竞争。”这句话不仅是线下零售的灵魂，同样也适用于电商行业，而对于电商供应链前端的采购平台，其重要性就更为突出。

伴随社会经济、科技的快速发展和市场环境的变化，生产供应显著增加，市场由生产导向型转变为需求导向型的市场环境，需求多样性逐渐显现出来，尤其对于网购消费者，其需求的不确定性和多样性更为突出。随着互联网科技的飞速发展，京东商城供应链环境下的采购管理有何变化？如何通过合适的供应商，以更低的价格、在精准的时间、按照优质的质量标准、获得恰当数量的产品用以满足消费者的需求？这一切都依赖于数据分析，大数据驱动着整个京东商城的供应链系统。

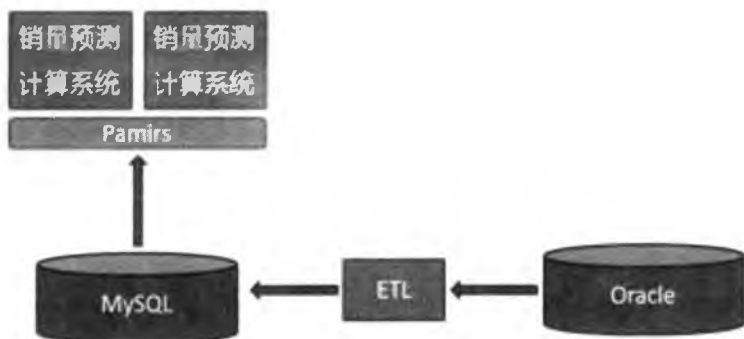




采销管理模块系统

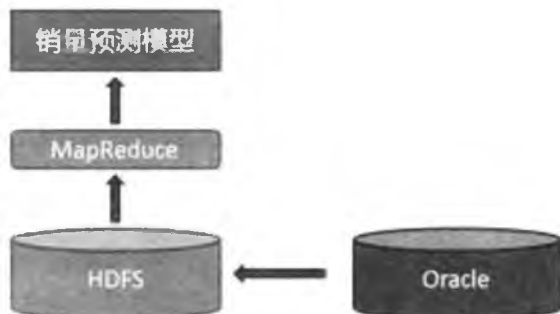
销量预测作为供应链的源头系统，它的准确率直接影响到供应链下游的自动补货、调拨内配和库存健康系统。利用大数据进行计算的销量预测模型便是系统的智慧结晶。那么怎样才能在海量的数据中提取出自己需要的数据，然后基于这些数据更精确地预测出未来的销量呢？2011年6月的某天，一份需求文档传递给了所有的小组成员，该文档的内容就是销量预测系统和自动补货系统的项目原型，提出了根据历史销量数据构建销量预测模型。在经过数据筛选、预测模型构建和预测模型开发与循环测试过程后，系统最终按时上线使用。

刚开始设计系统的时候，销量预测系统就在考虑大数据与计算性能的问题。当时采用了开源的 Pamirs 分布式框架：两台应用服务器在数据库中注册模值，不断从 Task 表中读取模值对应的任务。这个框架的好处是假如两台应用服务器不够用了，可以申请更多的服务器来提升计算性能。但开发的时候就发现主要时间开销都耗在数据读取上了。因为当时京东有上百万的商品，每个商品都要保留历史上所有日期的销量数据。一开始按照 SKU+配送中心+时间的主键保存数据，数据量很快就接近了亿级，现有的 MySQL 根本撑不住这样的数据量。后来重新设计了 MySQL 的 Task 表，让每个 SKU 都有一个记录历史销量的大字段，这样数据量就降为了百万级别。



基于 Pamirs 与 MySQL 的销量预测系统架构

2012 年京东的成长速度超过了所有人的预料，最直接的体现就是 SKU 数量从 2011 年的一百多万膨胀到 2012 年的 500 万。这个时候的 MySQL 支持销量预测系统已经显得非常吃力了，虽然研发人员不断对系统代码进行优化，但系统的计算时间仍然在不断增加。2012 年 6 月的时候，每天计算需要 6 个小时以上。系统架构的进步其实都是被环境逼迫出来的。同年 7 月，研发人员决定将销量预测系统向公司的私有云迁移。每天将 Oracle 中的数据同步到 HDFS，把增量数据合并到全量数据文本中（每一行数据为一个 SKU 与它的所有历史销量，避免了对 MySQL 的频繁读写），通过 MapReduce 调用预测模型完成计算。迁移后效果不错，每天预测计算时间压缩到了一个小时以内。这一年里，图书部门不断向我们提出改进建议。老刘的数据团队对预测模型做出了很多优化。模型加入了促销类型和库存状态，增加了大单过滤、极值平滑、短期促销过滤等。

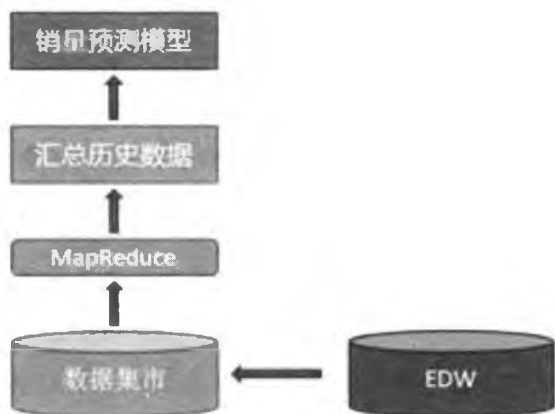


基于 Hadoop 的销量预测架构

2013 年，销量预测系统的研发团队承担起了数据与模型研究的工作。这一年

销量预测系统迎来了一个好消息：京东的数据集市投入使用了。研发团队在该集群上重新开发了销量预测的历史数据过程，获得了更准确、更稳定的历史数据。更重要的是，研发团队开始筹备新的预测模型。这一年中研发团队有很多新同事加入，他们都很擅长销量预测的建模工作。下半年，销量预测系统的工作重点在重建历史数据与预测模型上。研发部立项了目标为支持补货自动化的 Autopo 项目。但数据和模型两项工作纠缠在一起，而且只许成功不许失败（Oracle 当时正在计划下线），导致项目的难度很高。

每次例会，项目的压力都让团队成员情绪低落。4 个月的黑暗过后迎来了光明。2013 年 11 月底 Autopo 项目成功结项，数据集中建立了完整的历史数据体系，下线了预测效果不稳定的 Arima 模型，上线了更准确的价格模型、季节性模型和决策树模型。通过半年的时间，销量预测的准确率提升了 20%。从这个时候开始，研发团队才对销量预测的准确率有了控制权。

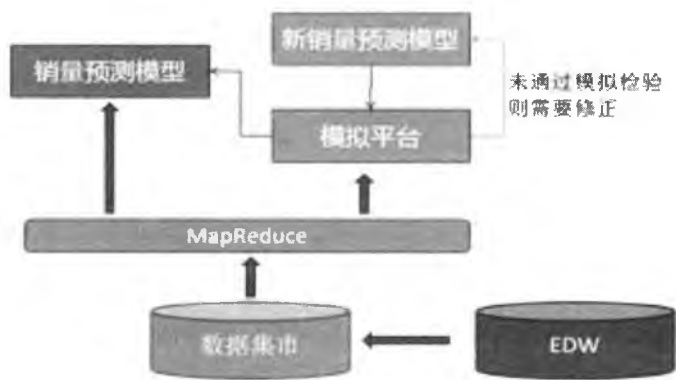


基于数据集市的销量预测架构

2014 年，销量预测的工作重心从以往的大数据性能优化上转移到了大数据驱动业务生产中。研发人员在建模的工作上越走越顺，对建模的热情也日益高涨。每个小伙伴都透露出积极投入建模工作的强烈意愿。2014 年 3 月，建立了预测模型的实验平台，设计出来各种模型都会在实验平台中模拟一次，看看对准确率的影响。以往所有的模型优化和系统改进都无法提前确认效果。经常出现的情况就是系统根据需求上线了一个修改或者微调了模型参数，第二天自动补货系统的补货数量翻了好几倍。

预测模拟平台是一个里程碑，因为它让销量预测的模型变为看得见的智慧，

也让销量预测系统在大数据的驱动下可以不断提高预测准确率。销量预测现在还在继续尝试设计新的模型。同年6月建立的促销模型可以计算出每个商品的价格与销量关系，支持将采销人员建立的促销计划转换为预测销量。一方面，销量预测系统将在促销期间提供更准确的预测销量，另一方面，可以通过促销模型来帮助公司制作出更有效率的促销计划，将智慧传播到促销决策的领域中，将大数据转换为看得见的智慧。



加入模拟平台后的销量预测架构

未来京东的销量预测系统将种下更多智慧之种，让供应链的智慧未来绽放得更加绚丽。

销量预测基本逻辑

JD.COM 京东



京东数据集市

以下是京东商城采购平台大事件：

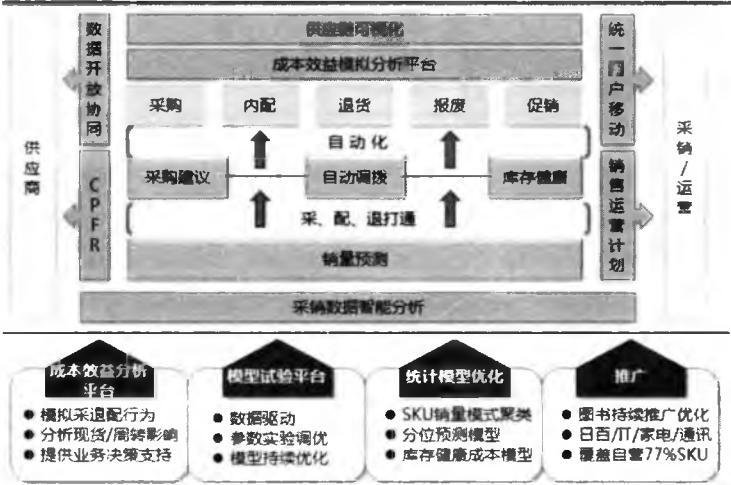
- 1. 2010 年 10 月，DOC 系统建立完成，主要适用于 3C 采销活动中的配件和五金产品；
- 2. 2011 年 9 月，为支持图书部门采购工作，建立了销量预测系统和自动补货系统；



采购建议系统——自动补货系统

销量预测自动补货库存健康系统简介

JD.COM 京东



自动补货库存健康系统

3. 2012 年 8 月, EPT-BIP 项目成立, BIP (采销集成平台) 为支持 EPT (Export) 海外销售业务, 对系统进行改造, 完成针对售向海外商品 (定制/非定制) 的采购工作;
4. 2012 年 9 月, BIP 系统上线, 一期实现采购员生成采购单及管理采购单相关功能;
5. 2012 年 11 月, 一键创建采购单功能实现将系统自动计算出的采购建议一键完成采购单创建工作, 该功能的完成预示着我们将向采购流程自动化又迈进了坚实的一步;
6. 2013 年 5 月, 实现了对进口商品的采购结算业务模式, 该项目扩展了京东对海外商品采购的业务线;
7. 2013 年 8 月, 自动补货系统支持了无库存销售客单自动产生采购单。该项目支持了京东自营商品的无库存销售运作, 提升销售;
8. 2013 年 9 月, 自动补货系统实现了全自动采购单生成功能, 同时, 为更好的完成采购单的审批工作, 提高采购单审批工作的严谨性及易用性, 对采购系统的审批流程进行改造, 提升了系统整体的实用性、安全性、灵活性等方面;
9. 2014 年 3 月, 零售平台的统一产品门户“采销空间”产品诞生;



采销工作空间示意图



采销空间

10. 2014 年 7 月，自动补货和销量预测系统已推广至全自营采销应用平台，同时，作为采销活动中最常用的生产系统“BIP 系统”（支持采购单创建），已成功运行 3 年半。



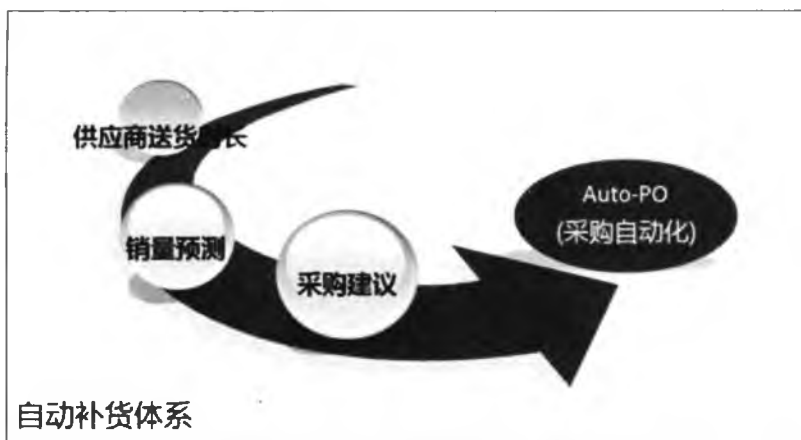
BIP 系统

# 自动补货

自动补货是与销量预测同一时期建设的系统，是销量预测的孪生兄弟，业务方最终需要的是系统能够帮助他们进行采购补货，销量预测与自动补货的最终结

果都是通过自动补货来体现。因此，相比销量预测的神秘和高大上，自动补货更需要贴近业务。

自动补货的完整概念包括采购计划和采购执行两部分，这里的自动补货更偏重采购计划，而采购执行是放在采购单系统（BIP）中的。什么是采购计划呢？理解这个概念，我们可以换一个思路，采购单是一个比较容易理解的具体的业务单据，其主要内容包括要采购的 SKU，每个 SKU 要采购的数量，以及供应商信息。现在来设想一下，采购单的各项数据是怎么来的？拍脑袋显然是不行的。这里一定有一个计算的过程，需要结合各种因素来最终确定采购单的数据，这个过程就是采购计划。简单来说，采购计划的职能就是确定什么时候采（采购时机）、采多少（采购量），以及从哪采（选择供应商）。而执行系统则负责采购单审核、下发库房、下发供应商、收货等流程的执行。



自动补货体系成员

## 准备、调研

2011 年上半年，对自动补货需求最强烈的业务部门是图书部门。因为当时图书部门的 SKU 数已超过 100 万，靠人工来补货已经是不可能的任务了。在这样的背景下，迎来了自动补货系统落地。

当时开发经理尚强带领的自动补货团队，由架构师建奇、开发人员老陈、老杨、媛媛和产品经理阿伟等成员组成。项目准备、调研就由这些团队成员与图书业务部门的张媛和老范等人共同进行。

美国网络零售中的领导者亚马逊和实体零售巨无霸沃尔玛都已经有了成熟的



自动补货应用案例。特别是亚马逊，智能化、自动化程度已经非常高，大部分采购行为已经是系统驱动。因此，亚马逊的自动补货自然就成为我们学习和参考的对象。但是自动补货是亚马逊供应链核心系统之一，没有任何公开资料，研发人员也全部都在美国本土，直接交流请教几乎不可能。我们的研发人员和业务部门充分展示了京东精神，舍我其谁，迎难而上。一直在做 Java 架构设计和代码开发的建奇同学，捡起了在大学时代就已尘封的概率论与数理统计，研究起概率分布和建模。做自动补货为什么要研究概率？补货是为满足未来的销售去补的，未来的销量是个不确定的东西，供应商多长时间能送到也是不确定的。对于不确定的事情，自然需要用概率来描述和处理了。

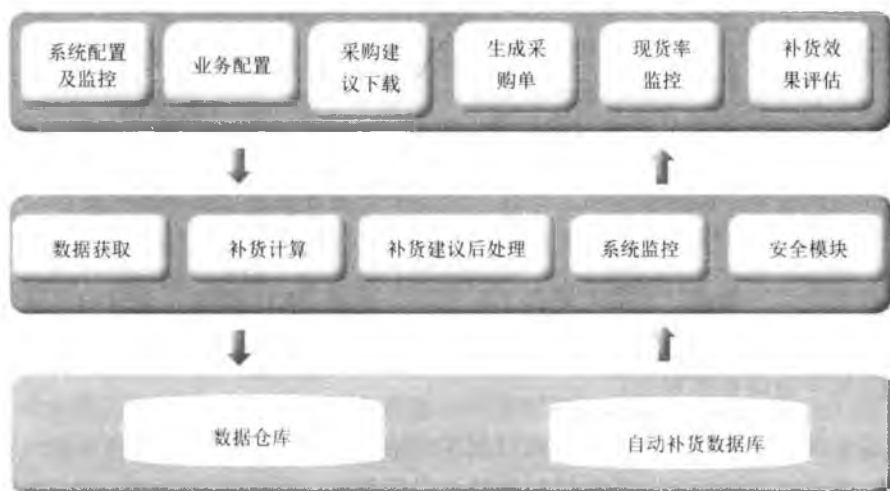
经过与业务部门多次沟通讨论，结合供应链业界在采购补货上的理论模型和实践效果，尚强、建奇等最终完成需求文档的撰写。与一般需求文档不同的是，除了系统功能需求的描述，文档中有大量数学公式、概率、图表、参数和数据的内容，详细描述了对销量预测数据、供应商送货时长数据、安全库存等数据的计算和使用方法。后来的小伙伴们在学习了解自动补货系统逻辑时，都要以看懂这份文档作为真正了解自动补货的基础。

## 设计、开发

前面需求调研和对销量预测、供应商送货时长和安全库存等数据处理和计算的概率分布模型建立和验证上花了较多的时间，留给设计和开发的时间就非常有限了。图书 SKU 量超过百万，北上广成武沈 6 个图书仓，一次出来的采购建议数据量就是千万级的，再加上销量预测、供应商送货时长等需要存储下来的中间数据，也都是千万级别的。计算和存储架构必须要考虑到这个数据量。

面对每天千万级别的数据量，分布式和分库分表就是必然的选择了。在分布式架构上，销量预测系统选择了基于 Pamirs 中间件的分布式调度架构，而自动补货系统采取了更简单的做法——直接按 SKU ID 取模的方式将数据分散到多个计算节点上（这一方式确实存在较多问题，稍后分解）。在数据存储上，考虑到补货建议相对于不补货建议的使用频率更高且数据量较小，就采取了补货建议、不补货建议分表的存储方案。事后也证明这个方案虽然简单，但满足了图书部门在查询性能上的需求。对于销量预测数据、供应商送货时长数据等中间数据的存储上，因为只是在排查问题时会使用到，就使用了 MySQL 大字段来存储。

这些设计方案在现在看来是比较初级，在互联网公司要讲求快，先出来东西远远重要于纸上谈兵。事实也证明，能满足业务需要的技术才是好技术。



自动补货架构示意图

## 上线、推广

自动补货项目上线后，给图书部门的采购补货工作带来了极大的便利。面对超过百万且不断增长的 SKU 规模，图书业务部门专门从事采购工作的人员非但没有增加，反而还在减少——更多的人员可投入到营销、发展优质供应商、合同/供应商谈判等更高附加值的工作中去。

除图书部门之外，2012 年上半年，自动补货开始逐步推广到各个品类，先是汽车用品和手机配件，然后是 IT 部门品类，至 2012 年年底，日用百货的一些品类和小家电部门的五金家装都陆续接入自动补货。

随着推广范围的扩大，各个部门也在原有的基础上根据各自的业务特点，提出了很多特定的需求和优化。例如，对于补货业务模型，从最初的一种，已经增加到了两种；补货参数的设置，已经从最初到结构和 Band 维度，丰富到品类、供应商、品牌维度，甚至细化到了 SKU 粒度的设置。

到 2013 年年底，CMO 办公室提出要在全自营范围内推广应用自动补货系统。自动补货系统也拉开了全品类推广的序幕。

在自动补货上线后的很长一段时间里，采购业务方对系统的使用方式一直是，从自动补货系统下单采购建议数据，review 并对需要的数据进行调整，再上传到采购单系统（BIP）形成真正的采购单。也就是说自动补货提供的服务并不是完全的自动化下采购单，更多的是向业务方提供采购建议，帮助业务人员做出及时

的合理的采购决策。

同时自动补货这个大而全的名称，也着实吓退了很多业务人员。一听到自动补货，业务一开始以为是系统就自动下采购单了。我下周要做促销，得多备货；我跟供应商刚谈好一个便宜的采购价格和采购批量；我这个商品要做大宗采购、要买断方式采购；供应商有固定的排产计划……种种疑虑让业务方听到自动补货这个名字，一股担心就涌上心头。这给我们的系统推广工作也带来了一定的困难。

我们对系统的推广，并不是马上、也不是全部 SKU 都进入全自动化补货。毕竟，各种补货参数不是一开始就能够确定下来的，需要在业务生产过程中根据实际效果不断调整到最优。

鉴于此，2012 年年中，开发经理刘术成给自动补货建议了一个别名——采购建议。由于我们最初向业务提供的是建议采购数据，所以叫采购建议也是比较合理的。在这之后，自动补货才有了采购建议这个名字。如果你看到采购建议这个名字，那它实际上就是指自动补货。

## 自动化采购下单

前面已提到自动补货上线之初，并没有急于去做自动化下采购单。2012 年下半年，自动化下单开始实施。在自动化下单上，仍然是分两步走的。

首先是实现一键下单，也就是采销人员在自动补货系统中，点击一个按钮就可以将相关的采购建议数据传送到采购单（BIP）系统生产真正的采购单并执行后续采购单流程。之所以还需要业务人员点击按钮这个动作，是业务人员需要对自动补货系统数据进行 review。这个中间步骤也叫“一键下单”。

到 2013 年 7 月，图书的 EDI 采购单实现了完整的自动化下单。自动补货计算好采购数据后，无须任何人工干预，直接进入采购单系统自动化生成采购单并自动化发送给供应商以及执行后续采购单流程。

## 架构升级

从 2012 年上半年开始，自动补货的推广范围逐渐从图书扩展到了 IT、通讯、日百、家电部门的众多品类。系统也增加了许多针对不同品类的需求和优化，在这个过程中，系统原有架构应对新增需求显得越来越吃力。同时，推广应用范围的扩大，对系统性能和稳定性的要求也越来越高，原系统架构的弊端逐渐显露。从各个方面来看，系统技术架构的升级迫在眉睫。

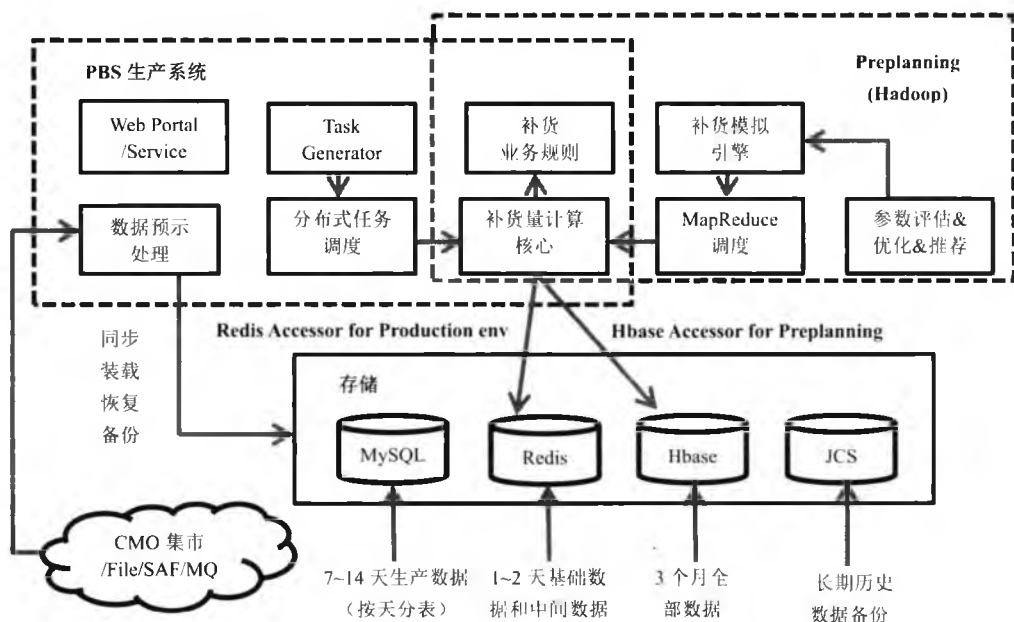
首先改造的就是业务模型。开发经理尚强提出了面向业务的采购计划概念，

将 SKU 按采购计划进行分组，每组可以根据不同的情况设置不同的补货参数，更有利于系统满足不同部门、品类业务人员的不同需求。同时有了采购计划这个层次，权限控制也更方便了。产品经理老吴将这个方案细化，阿斗、媛媛、老杨在 2012 年年中将自动补货采购计划版本开发完成并发布。这个改造上线后，受到业务部门的热烈欢迎。

最初的按机器数量取模划分任务的分布式调度框架，随着应用范围的扩展也逐渐暴露出了问题。比如，如果某台机器挂掉，其所负责的任务就丢失了，最终数据就是不完整的。已经有了采购计划这个概念，我们就将每个采购计划作为一个任务调度粒度。为了避免某台机器挂掉任务丢失，任务调度中加入了乐观锁机制，辅以监控重试，增加或者减少机器对任务调度都是透明的。后来为了增加并发性，我们将每个计算节点的实例增加一倍，但在代码上无须做任何改动。

在新的架构设计过程中，老杨敏锐地发现，各个采购计划的数据量是严重不平衡的。例如图书的一次采购计划，就有超过百万的 SKU，而其他品类的采购计划 SKU 数量一般在 5000 到 20000。SKU 分布的这种极度不平衡性，会让我们的基于采购计划粒度的调度架构退化为线性顺序模式，完全失去并发性，计算耗时现象严重。解决这种数据分布不均衡问题，老杨很自然地想到了虚拟化技术，一致性 Hash 不就是这么做的嘛。因此，虚拟采购计划的概念被提出来，把 SKU 量超大的采购计划切分成小的虚拟采购计划，分布式调度时以虚拟采购计划作为基本任务分派粒度。

而 MySQL 数据库在这个时候也出现了承受不了每天增加超过 50G 数据量的压力问题，DBA 找我们清理数据库的频度越来越高，从半年一次到三月一次到一月一次直到每周一次。线上历史数据能保留的时间段也越来越短，最后只能保留一周的历史数据。而作为一个数据驱动的系统，需要对大量的历史数据进行分析，以调整优化系统补货效果。这就要求历史数据要尽可能地保留下来。考虑到还要建立补货模拟平台，我们没有将历史数据转入数据仓库，而是选择 Hbase 作为历史数据存储介质。这样历史数据就自动转存到 Hbase 里，近期数据会留在 MySQL 中。同时，原有的按补货和不补货建议的分表也更新为了按计算日期和采购计划分表，以满足日百、IT、家电等部门需要查询补货建议的需求，对于业务部门来说，不补货建议也是有价值的数



基于云计算的自动补货架构图

## 补货模拟平台建立

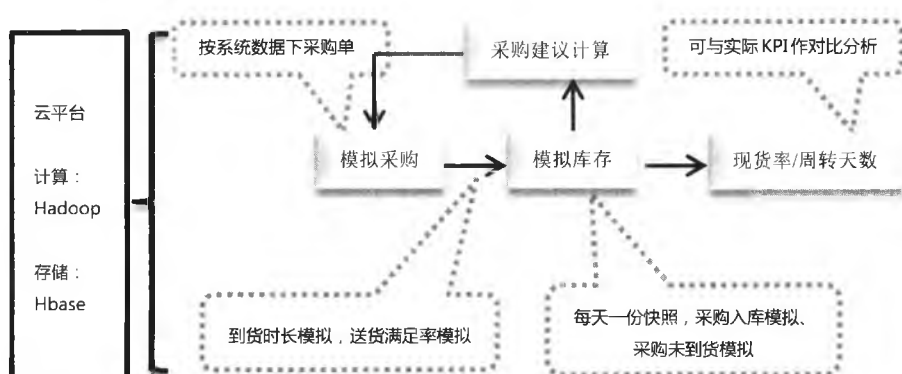
现货率和周转是供应链的核心指标，提高现货率、降低周转天数也是自动补货系统的目标。补多了、补少了、补早了、补晚了都会对现货率和周转产生不良影响。怎样才能评估自动系统数据的合理性呢？怎样才能评估自动系统数据对于现货率和周转的影响呢？如果自动补货数据全部得到使用，就可以根据事后的现货率和周转的情况评估自动补货数据的合理性和效果。但是，不能充分说明系统数据的合理性和效果之前，业务方不会贸然大量采用系统建议的补货数据。这就让系统的推广面临先有鸡还是先有蛋的问题了。

补货参数是需要调整和优化的，如果完全的补货参数调整和优化的过程放到实际生产中进行，效率会非常低下，各种干扰因素影响下，也难以达到预期效果。

我们是数据驱动的供应链，必然要基于大数据来解决这些问题。

2014 年上半年，基于大数据技术的补货模拟平台启动。补货模拟平台运行于 Hadoop 集群中，会使用历史半年至一年的全部历史数据，包括库存、出/入库量、销量、采购。模拟完全按照采购建议的数据进行采购生产，推演每一天的库存变化形成模拟库存，并使用模拟库存循环计算新的采购建议量，进而将从模拟库存

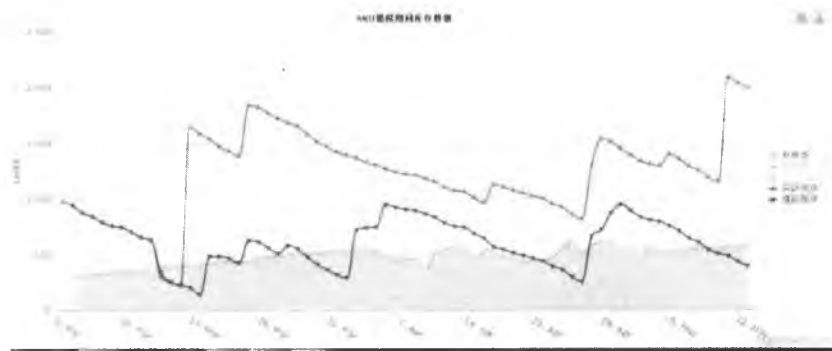
计算到的现货率和周转天数与实际的现货率和周转天数进行对比分析，定量评估系统建议数据的合理性和业务运营的优化程度。



补货模拟过程示意图

模拟的结果是令人激动的，系统数据如果被全部采纳，在不降低现货率的情况下可以减少数亿元的库存成本。

模拟过程中库存的变化情况也能充分说明系统建议数据对业务运营的优化效果。如下图为某商品在模拟过程中的库存变化，模拟情况（下面的线）显示只需要实际（上面的线）一半的库存量就能够达到同样的现货效果。



补货模拟效果示意图

如今，补货模拟平台已经开始在系统推广和补货参数优化上发挥关键作用。全库存控制行为的模拟平台也已开始建设，由于揭示每一种库存优化行为、每一次库存控制行为、新业务的变更，在实施之前，我们就能够定量评估出其所带来的效果和价值。这就是大数据驱动供应链的魅力。

## 数据罗盘

2014年6月16号的上午9点，六月的天，说热就热，作为一名冰箱供应商，这个时候显得尤其兴奋，同时却也要更加谨慎。因为既要抓住高温带来的购买高潮，又要时刻保证库存和备货。和平常一样，老郭坐到了办公桌前，打开电脑，习惯性的点击了“京东罗盘”，先查看了一下商品的库存情况，还好没有预警，我们有足够的备货应对即将来临的购买高潮。再看一下最近的销售和流量情况，咦？怎么回事？盛夏高温销量应该上升的，为什么下降了？尤其是本应该热销的“海尔爆款”。



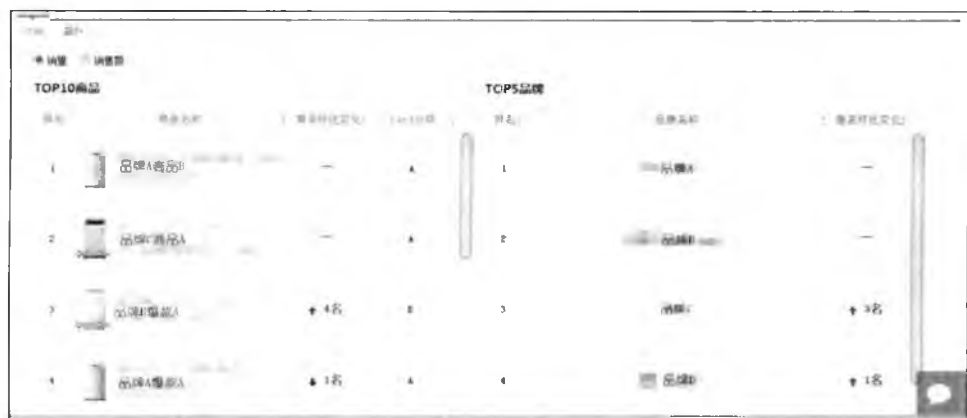
数据罗盘之基础分析

老郭急忙打开行业分析模块，查看了一下空调行业整体的行情变化，原来整个行业的销售额没有明显的下降，还是一个比较平稳上升的趋势。



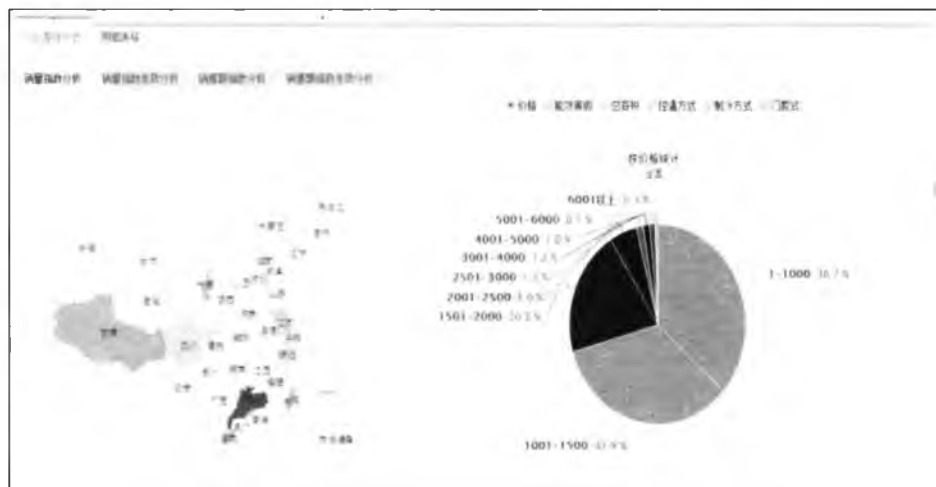
数据罗盘之行业分析

通过热销分析，同样发现“海尔”商品 1032689 排名下降了 3 位？这到底是怎么回事呢？



数据罗盘之热销分析

随后老郭决定了解一下“海尔”现阶段的市场份额，品牌认知度以及消费者特征等信息，在市场分析模块，我可以看到整个冰箱市场，什么样的款式卖得好、什么价格段卖得好、在哪里卖得最好以及哪里的潜力比较大等信息，通过区域地图颜色的渲染以及关注度和搜索份额、销量等多维度的交叉比较，老郭发现相对于“美的”“TCL”，“格力”等品牌，“海尔”在“广东”卖得最好，而在“浙江”的市场潜力最大（搜索度和关注度高，销量少），“浙江”将成为潜在市场。



数据罗盘之区域分析

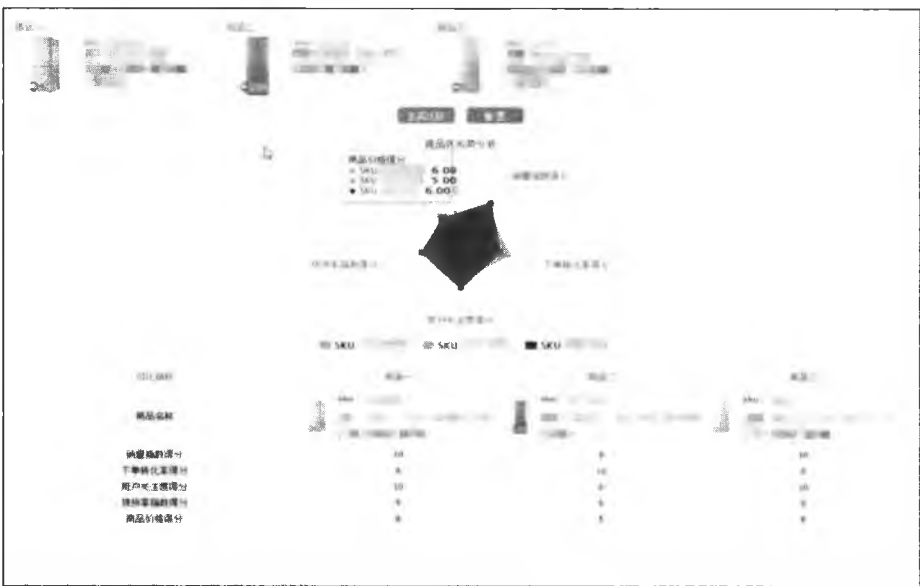


针对本应该热销却排名比较靠后的“海尔”商品 1032689，老郭点击了流量分析，可以看到，商品 1077304、480730 等对商品 1032689 分流量最大。



数据罗盘之流量分析

选取这两个商品，通过竞品的优劣势分析发现，商品 1032689 在用户关注度维度处于劣势，同时关注竞品的用户也成了老郭的潜在用户。



数据罗盘之竞品分析

关注到上述信息以后，老郭把各个模块的数据进行了一个较系统深入地比较分析，针对分析结果做了一份营销策划，目的在于增加品牌影响力和用户黏度，从而增加销售额。首先，老郭决定参加从 2014 年 6 月 17 日到 19 日为期 3 天的 618 大促，并且通过了解页面流量的来源，进行针对性的广告位投放；其次，老郭需要在



市场状况，老郭发现经过促销活动以及针对性宣传活动，在潜力市场城市浙江，品牌认知度有所提升，销量也有了较大的增长。

面对瞬息万变的 market 格局、不断变化的客户需求，京东罗盘可以更好地辅助更多像老郭一样的供应商快速获悉行业的发展趋势，了解市场的竞争强度，洞察消费者行为，从而可以让供应商及时有效地采取相应措施。数据让商业决策变得更加简单。（作者/胡浩、谢蔚、马斗、张国利、郭建华、董浩、李帅、张蕾）

# 19

## 蝴蝶效应——创新孵化

### 创新是一种价值观

京东的“创新”由来已久，可以说从老刘创业时就开始了，当年的老刘首创电商的“荷兰式拍卖”，而那时的京东还没有夺宝岛的拍卖和一口价。老刘凭借这一小小的产品创新吸引了众多眼球，为京东带来了不小的流量。也就是从那时开始，创新成为了一种基因；出现在了每个京东人的身体里。

2013 年年初，京东高管齐聚集团总部北辰 533 会议室，决定对企业核心价值观进行全面梳理。仅几天的时间，京东新版核心价值观出炉，“创新”作为新的价值观被纳入其中。而这一天也标志着京东开始为自己的基业长青做好了准备。

自 2004 年京东涉足电子商务领域，我们的业务就以每年 200% 的跨越式速度向前飞奔。在这个过程中，京东技术一直扮演着支撑者的角色，保障着上层业务的有效开展，支持着每年上百亿的销售额。到 2013 年为止，京东技术已走过 9 个年头，618 店庆的大流量已经不再是系统的最大挑战。京东技术该走向何方？它在公司的发展过程中又该扮演怎样的角色？

就在“创新”被纳入京东核心价值观后不久，Donny 为京东技术定下了“三步走”的战略：Follow-Lead-Drive。而这“三步走”战略的核心就是技术创新。京东技术只有通过创新才能更好地支撑业务，也只有通过创新才能真正实现技术

驱动业务向前发展的美好愿景。京东技术创新之路从此起航。

对于技术人员来说，创新似乎并不陌生。他们每天的工作都是在对系统和产品进行不断的优化和改进，而这些工作在某种意义上即为创新。值得思考的是，技术团队能否做出更大的创新，甚至改变世界呢？如何调动起技术人员的积极性，让其充满热情地投入到创新工作中，确实是创新面临的实际问题。经过几番思考与讨论，京东产品论坛应运而生。

## “京东网站未来五年畅想”

自从京东产品论坛的想法提出后，第一期活动就进入了筹备阶段。Zoey 的团队承接了全部的活动组织工作。对于他们来说，举办这样的活动也是第一次，既紧张又让人激动。经过多番讨论，老大们决定第一期活动围绕京东网站改版展开，主题为“京东网站未来五年畅想”。

这个主题被抛出去后，立刻激起了京东所有产品经理们的兴趣，大家都兴奋不已。一是京东首次举办这样大规模的产品创新活动，二是此次创新活动将为产品经理提供一个充分展示实力和想象力的舞台。而且在活动中，产品经理们不必顾虑太多技术实现的问题，只要大胆畅想即可。

本次活动自 2013 年 4 月初正式启动，历时一个月，按照计划，各团队的网站创新产品将在 2013 年 5 月 4 日正式对外公布，进行同台 PK。本次活动的冠军团队，将有机会实现自己对网站的设想，未来京东商城主页的改版将参照此方案执行。这也是让产品经理们兴奋的地方，自己的方案可以改变几千万用户的购物体验，这样的机会恐怕一辈子也赶不上几次。大部分产品经理按照部门进行了组队，也有几名研发人员参与其中，报名结束的时候有 14 支队伍共六十余人。各参赛队在比赛的过程中，充分发挥团队成员的想象力。按照比赛的要求，参赛队不仅需要提供网站的设想，还要将 Demo 完成，以便最后的展示。各团队都在利用业余时间进行讨论，想设计一个完美的让人眼前一亮的作品。

时间飞快，一个月的设计时间转瞬即逝，2013年5月4日就在眼前。各参赛队的产品设计质量究竟如何，只有在大赛上才能见分晓。而对于组织方来说，最后的决赛才是最关键，我们一定要确保各项准备工作万无一失。Zoey的部署可谓是面面俱到，首先他选择了北辰一层员工休息室作为决赛现场，那里既宽敞明亮，又充满创新的气息，就连座椅都是五颜六色的，让人感觉非常舒服；在评委的选择方面，不仅邀请到了技术团队的多位高管，同时力邀了京东战略部总监加入评审团，另外还从公司外部邀请到NOP品牌创始人刘爽先生担任特邀嘉宾。

2013年5月4日，活动如期举行。即使当天是周六，一层休息室还是站无虚席。活动宣传已经从研发扩大到集团总部的所有部门，无论是不是技术体系的员工，大家都对我们网站未来的样子充满了期待。就连老刘也不例外，他原本在决赛当日有一整天的会议，但还是在开会的间隙特意到会场观看选手们的分享。老刘到场的时候，大部分员工都没有发现，因为他一直站在会场的最后。说来也巧，老刘当时观看的作品最后获得了大赛的一等奖。当时的分享人汗青也获得了老刘的表扬。



《京东是一种生活方式》演讲人汗青

本次大赛各个参赛队带来的产品创意可谓是各有千秋。UED团队的《京东是一种生活方式》，让“京东”成为了客户生活的小帮手；仓储物流产品团队的《京东物流产品畅想》，将配送一体机、社区集市及售后客服开放平台完美结合，大幅度提升了客户的购物体验；还有《我与自我的距离》、《3D购》等产品均获得了评委的一致好评。

通过京东产品论坛第一期活动，不仅产生了很多优秀的创新作品，还有力地向全体技术人员传递了创新的信号，激发了产品经理们的创新热情。

第一期活动结束后的第三个月，京东“青春”正式上线。而这个产品的原型正是第一期产品论坛第一名的作品《京东是一种生活方式》。

创新的想法能够落地，并开花结果，这是对创新人才最大的鼓励。无论产品最后的市场反响如何，这都是一次大胆的尝试，更是一次技术的飞跃。而“京东网站未来五年畅想”活动也成为京东技术创新史上的第一个里程碑。



京东产品论坛第一期合影

## 孵化器诞生

2009年9月创新工场在北京建立，它是一家为创业者提供全方位创业培训的投资机构。创新工场通过针对早期创业者需求的资金、商业、技术、市场、人力、

法律、培训等提供的一揽子服务，帮助早期阶段的创业公司顺利启动和快速成长。创新工场至今已帮助众多有创业梦想的年轻人走向成功。

大部分 80 后、90 后的年轻人已不会像父辈一样安于现状，他们更希望做自己喜欢的事，创造属于自己的未来。因此，一部分人放弃了稳定的工作，选择了创业。

对于京东这样一个年轻的企业来说，员工平均年龄还不到 30 岁。在这里，有无数年轻人渴望突破束缚，一展才华，尤其在京东技术团队更是如此。他们受过良好的文化教育，很多都是研究生。经过几年的工作积累，他们在细分的技术领域也都小有建树，一部分人就会产生创业的想法。如果这些年轻人离开京东，对于京东来说，无疑是重大的损失。借用《天下无贼》中黎叔的一句话：“21 世纪什么最贵——人才。”所以，如何保留这些充满激情的年轻人就成为一个关键问题。

另一方面，老刘早已将京东定义为一家技术驱动的企业。而技术究竟如何能驱动京东的业务不断向前发展，答案只有一个——靠创新。创新的不仅是技术和业务，更是通过技术手段解决消费者面临的实际问题。只有这样才能真正实现业务的拓展。

2013 年双 11 后，Donny 与老刘在美国见面，为京东技术规划未来。Donny 回国后，提到的第一件事就是筹备京东孵化器，具体工作由创新组 Clement 负责。接到这个任务后，Clement 激动不已。他曾经与朋友一起创办过一家公司，对创业充满热情，但外部创业的残酷让他懂得环境的重要性，如今获得了更多的资源和指导，创业成功的概率就会大大增加。

京东孵化器会是怎样的一个组织，它该如何运转。Clement 只能通过与 Donny 的沟通和查阅相关资料获得更多的信息。资料显示，在全球顶尖 IT 企业中思科是较早尝试内部创业机制的。《浪潮之巅》里对思科的内部创业有这样的描述：“如果公司里有人愿意自己创业，公司又觉得他们做的东西是好东西，就让他们留在公司内部创业而不要到外面去折腾，而思科会作为投资者而不再是管理者来对待这些创业的人。一旦这些小公司成功了，思科有优先权把他们买回来，思科的地盘就得到扩大。而这些独立的小公司的创办者和员工，又可以得到很高的回报。这样本来想离开思科出去创业的人也就不麻烦了，接着上自己的班，只是名义上换了一家公司。”这样的模式是否适合京东呢？Clement 带着疑问，与相关部门的领导进行了沟通。沟通后有两个问题不容忽视：第一就是中美文化差异，国内企业和员工相对保守，担心创业风险太大，收益无法保证，所以绝大部分人对创



业并不看好；第二就是企业内部规章制度繁琐，相关制度的建立受到限制。

只有针对京东现状制定针对性的创新机制才能保证孵化器有效运作起来。在工作中最重要的两个关键点就是人员安排与项目管理。人员安排方面，进入孵化器的员工原有薪资不变，绩效成绩统一由项目负责人评定。这样可以促进更多员工进入孵化器，而不必为生活发愁。项目管理方面，孵化器采用项目 CEO 制，项目 CEO 负责整个项目的管理，人财物统一把控，这一点与创业团队一样。这样的机制给团队更多的自主权，也让项目的发起人真正按照自己的想法去做事。孵化器的所有机制设计都以提升员工积极性为原则，保证创业团队的权益，同时尽最大可能解除创业团队的后顾之忧。

经过两周的意见征集与不断修改，《京东 R&D 创业孵化器计划》终于出炉。整个方案涵盖从 Idea 产生到项目结束的所有规定。它就像一条通往未来的路，创业团队就是在这条路上开车的人，究竟最后是否能成功抵达终点，全看团队的驾驶技术。而这条路上，孵化器就是加油站，尽可能提供一切的补给。

计划发布后的一个月内，创新组收到了京东研发体系员工提交的 20 多份商业计划书。商业计划书涉足的领域广泛，O2O、公益、游戏等。值得思考的是，究竟如何对这些商业计划进行筛选呢？哪些团队会成为幸运儿，首批入驻孵化器？这一切似乎都是未知数。Clement 与创业团队同样充满期待。

## 4 个“蛋”

第一批商业计划书收集结束后，根据孵化器相关规定，每一个想要进入评审阶段的项目都要在一级部门的 Leader 里面挑选一位作为自己项目的导师。导师制的建立是为了帮助创业者们更好地完善他们的商业计划书，同时也有一个初步筛选的过程。如果一个方案找不到一位支持者，默认这份方案是没有办法通过最终评审的。

经过导师们的筛选，一共有 6 个项目可以参加评审。评审会上，由京东研发体系高管组成投资委员会，采用投票制决定项目是否通过。因为平时与高管们接

触较多，Clement 对高管们所关心的项目重点也比较了解，在评审会开始前已经有针对性地与创业团队进行过沟通，期望他们能有更好的表现。有时候项目的好坏其实差别不大，都会存在一定风险，更关键的是看项目展示环节和团队的发挥。

参加评审的项目有母婴类、汽车类及医药类等，每一个项目的侧重点都不太一样，创业团队的表现也千差万别。汽车项目的时涛在展示时就比较紧张，回答评委提问的时候也抓不到重点。而医药类项目在展示时就相对老练，高管们能问到的问题，他们都有所准备，从容不迫。最后，经过高管们讨论决定，6 个项目中的 4 个通过。

当 Clement 将最终通过的消息告诉创业团队时，他们都欣喜不已。终于可以开始做自己的项目了，下一步他们就需要真的像创业团队一样去工作。在前面等待他们的将是更加艰难的创业之旅。

每个团队都会面临大大小小的考验。项目通过后，木桃团队就经历了创始成员离职的风波。创始团队一共就两个人，现在就只剩一个人。怎么办？是放弃还是继续？创始人忠良与 Clement 表了态，即使一个人他也想做下去。那么接下来，重要的就是找人来。经过与 Donny 的讨论，决定由成都研究院担任开发工作。但是问题又接踵而至，成都团队为了保证产品的质量，希望不仅仅扮演开发者的角色，还要更深入地介入产品设计。成都团队的初衷是好的，但却带来了一些矛盾：一位创业者能让其他人随意干涉自己的产品吗？这产品就好比自己的孩子一样，有太多的感情在里面。如果不放手的话，只能另外组建团队进行开发。Clement 在这件事上的态度是很坚决的，宁可暂停项目也不能采纳成都团队的建议。因为这样做将极大地挫败内部创业者的热情，对于他们来说，能做自己的产品比能得到回报更重要。所以迫不得已，忠良只能另外寻找团队。刚开始开发的一两个月，他自己找到了两个兼职的同事一起做，但由于对开发语言不是特别熟悉，进展较慢。两个月后当我们 Review 他的产品时，Donny 果断决定建设专门的团队支持忠良的开发。历尽波折，木桃团队终于正式成立了一个 7 人的小团队。

除了木桃外，其他 3 个团队也有各自的难题。不过当他们遇到难题时，高层总会想尽办法给予支持。他们自己也会用尽所有办法去克服困难，在这个过程中，孵化器能给他们提供的也只是杯水车薪，有太多的业务问题需要他们自己解决。创业的路上，谁都是没白天没黑夜地工作，要的只有一个东西——成功。

孵化器的作用是什么？进入孵化器就一定能成功吗？孵化器能帮助创业项目提升零点零几的成功率已经是起了很大的作用了。在这里要成功也一样要靠天时、地利、人和。

京东孵化器计划为内部创业者们提供了一个平台。在这个平台上，他们可以做更大的创新项目，他们也同时面临着更大的风险。创业，是为勇敢者准备的游戏。

## “微创新奖”

对于京东技术的创新管理方法，创新组的成员们一直在探索，也在不断学习国内外著名互联网公司的先进经验。

五月的深圳已是三十多度的天气，Clement 借着与 ECC 融合的机会拜访腾讯总部。与腾讯大讲堂的同事们共同探讨双方的业务，这是一次难得的学习机会。在互联网领域，BAT 三足鼎立，值得京东学习的还有很多。

在深入交流的过程中，Clement 特别注意到腾讯大讲堂每月会举办一次微创新奖评选。微创新奖选中的产品可能非常小，甚至只是一个功能点，但这样的激励方式极大地鼓励了腾讯员工的创新热情。每个人不一定要做出什么伟大的创新产品，只要你在自己的工作中做出细小的改进，哪怕只是把用户体验提升一点点，你都可以申报微创新奖。它是一种鼓励，更是一种认可。

当 Clement 回到公司向上级汇报工作时，总监 Bill 也特别注意到了这一点，认为这项措施可以立刻在京东技术体系执行起来。它将帮助京东技术创新文化再上一个台阶。Bill 很快就向 Donny 做了汇报。Donny 指示先做方案，通过管理层会议评审后即可开始执行。

Clement 很兴奋，老大们这么爽快地同意了这个想法，两天之内《京东微创新奖》方案就发送到了 Bill 的邮箱。经过讨论与修改，管理层会议最终通过了提案。微创新奖从 6 月开始执行，每季度一次，每期总奖金额 50000。

第一期从宣传到收集作品，时间不过 3 周。在提报的过程中，也发现大家对微创新的定义并不是特别清楚，很多产品是很大的，需要耗费的人力和时间也较长。虽然严格上讲，这些产品不算在微创新的范畴，但是为了鼓励大家积极参与，第一期的微创新奖并没有严格限制，只要是最后通过评审的作品，我们都给予相

应的奖励。

我们认为精神激励永远大于物质。什么样的精神激励可以让获奖的同事感到无比光荣？我们制作了京东“微创新墙”。而这一点，其实也是从南洋理工大学学来的。在南洋理工大学的一所学院里，专门有一面墙展示着这个学院每年各专业成绩第一名同学的名字，这对于学生来说是无上的光荣。几十年后，当年的第一名依然在上面。而“微创新墙”也将成为京东技术创新的历史书，它将真实地记录这些微创新的英雄们，是他们对京东的技术创新做出了卓越贡献。

微创新奖对于京东来说就是九牛一毛，可能连一毛也算不上。但是它的意义却很重大，正是通过这样的方式促进了全员创新，无论你做出怎样的改进，你都会意识到为企业做出了贡献。它是一种期待，也是茶余饭后的谈资。希望它在京东技术人的心里种下一粒创新的种子。



微创新奖奖牌

## 创新是一种文化

2013 年年初, Clement 还记得第一次与 Donny 共进午餐时 Donny 说的话, “现在京东技术的创新没有专人负责, Clement 你来想办法, 争取把这块做起来。”当时的 Clement 其实也不知道该如何做, “创新”听起来毕竟还是一个比较虚的词语。而且怎样才算做得好, 那时也并没有特别清晰的概念。Clement 想, 大概要先从贴条幅入手, 这完全是到农村刷墙的做法。后来, 随着创新活动的举办、孵化器的成立、微创新奖的引入, 技术创新的地图才慢慢展开。

创新的管理是从培育、获取、选择、实施、收益到激励的完整闭环。只有这个闭环形成了, 创新才会源源不断。京东技术创新也正是遵循着这样的思路来开展的。京东产品论坛是获取创新的有效手段, 也在培育着一种氛围; 孵化器是独立的创新机构, 在这里有最自由的创新空间; 微创新奖是号召全民创新的推动器, 从微小的创新开始, 大的创新才会越来越多。

企业内部的创新氛围营造至关重要, 它就像育苗的过程。而培育也与企业文化、架构及制度有关, 京东在 2013 年就将创新列入核心价值观, 也是企业文化的重点。我们的架构也朝着小团队模式发展, 更加灵活, 反应更加迅速。制度和流程有时候是创新的巨大阻碍, 而京东技术团队正致力于砍掉所有影响效率的流程, 力争在效率与质量、效率与安全上取得平衡。

在创新的另一个关键环节——激励上, 京东技术团队每年投入几百万专项资金。大大小小的活动, 各式各样的产品与项目, 只要是有价值的创新, 都会去激励。更重要的是, 让所有技术人员在创新的过程中获得满足感、成就感和认同感。这些创新者们的名字也将会一直留在京东的史册上。

未来, 京东技术创新将立足全球, 在世界范围内寻找最强大脑, 这是创新的基础。最上层是京东的愿景——让生活变得简单快乐, 这是未来的创新方向。从上至下或是从下至上, 两种不同的创新方法, 都将在京东的平台上展现。最终, 创新也将走出京东技术, 走向京东的每一个角落, 形成创新的大文化。(作者/樊晨)

# 牛人

英雄惜英雄。在众多产品、技术、创新的背后，到底是哪些人在领导并实践着，本篇我们就把这些幕后英雄介绍给大家，他们才是这个电子商务世界真正的造物主。

来吧，一起跟牛人学习如何改变世界！

· 京东技术大咖



# 20 神盾局——京东技术 大咖

## 开拓者——邓正平

移动、智能硬件、智能云方向牛人

#是金子总会发光的，坚持就是胜利！#



在京东研发，有几个人是大家公认的传奇人物，其中一个就是邓正平。

2014 年年初，京东集团 CEO 刘强东召开媒体发布会，宣布京东未来 5 大战略，其中第一项就是移动。当时有记者问：“京东的移动起步是否晚了些，丧失了移动的第一张船票？”然而了解内情的人都知道，京东早在 2010 年便开始了对移动客户端开发的探索，而京东移动业务的第一任负责人，就是正平。

2010 年，正平从零开始组建京东移动团队。两年间，带领团队研发了基于 Android、iOS、Symbian、Kjava、Wp、Win8、TV 等平台的移动客户端和手机网站，在移动端实现了京东商城大部分实物及虚拟业务的交易。

由于当时大家对移动的认知不足，有人甚至质疑说：“谁会在手机那么小的屏幕上买东西？”当时，移动业务开展艰难。然而，到 2013 年，移动端用户量已达数千万，订单占比达 10%。正平带领团队实现了京东移动端的发展壮大，为京东移动战略奠定了坚实的基础。

2013 年年底，公司开始布局智能硬件。又是一个新领域的开拓，又一次从零开始，公司再次选择了正平——由正平担任京东智能云负责人，主导京东智能化业务技术研发。2014 年 6 月，正平率领团队成功打造了业内首创的智能云平台 and 京东超级 App。该业务致力于和各类硬件厂商、芯片解决方案提供商、应用服务提供商、第三方开发者等合作伙伴共同打造智能硬件领域的入口级平台，以及多方共赢的智能硬件生态链。目前，包括海尔、海信、美的、TCL、志高、华为、海康等行业巨头在内的数十家企业已经接入该平台，使其成为中国最具影响力的智能化物联网平台。

回首这些开拓的日子，正平说道：“加盟京东 4 年，见证了京东从简单的 .Net+SQLServer 系统变迁到主要基于 Java 的大规模分布式系统，见证了京东的研发部门从一百多人发展到现在数千人，见证了京东从一个 3C 垂直电商扩张为覆盖全品类 B2C 电商、C2C 拍拍、金融、物流、海外事业部等业务的互联网巨头，见证了京东从苏州街边上的一个小公司快速成长为一个市值超过 300 亿美元的上市公司。但是这一切都只是开始，京东的目标是要成为交易额过万亿的中国最大的电子商务公司，我们面临的技术挑战将会是空前巨大的，期待有更多的技术天才加入京东。”

作为一名开拓者，邓正平既有一线的软件开发和系统架构设计经验，又有多年的团队管理及业务成功开拓经验。在加入京东前，他先后在搜狐、中软等企业就职，丰富的工作经验和敏锐感，让他有常人所不能及的战略眼光和高度，更有敢为人先、不断创新的开拓精神，也就是京东价值观中强调的“创新”精神！

目前，正平继续肩负着为京东智能化物联网业务开疆拓土的使命。



## 百变神龙——杨思勇

POP 平台、移动、电子书、网站前端方向牛人

#战斗战斗，只做第一#



叫杨思勇百变神龙，是惊叹他在不断变化的工作上表现出的适应性和做出的成绩。看看思勇在京东的履历就知道此言不虚。加入京东后思勇首先带领团队向 Java 方向转移。用思勇自己的话说就是“接到了这个活，然后啥也不懂，幸好找了几个特别懂的人，然后一起学，一起弄，最后才搞出来的。”这几句诙谐的调侃背后是我们难以想象的艰辛和努力。

思勇初来京东时，京东总体研发水平落后，工具比较原始。团队要转向 Java，只好分头去学习先进技术，然后进行内部分享。也从那时起，分享的传统被保留下来。后来，当时一些好的分享和实践还被汇编起来，纳入京东大学的开发人员最佳实践课本。

2010 年，马松带领思勇展开 POP 开放平台的开发。POP 开放平台创建初期，需要把京东所有系统全部打通，并提供给第三方的牛人使用。当时正流行 SOA 解决方案，于是他们就跟订单、商品、售后等系统人员沟通，采取由他们写界面，直接调用老系统的接口来展示数据的方法。结果发现性能无法接受，页面浏览时间需要 3 秒以上才能展示出来，不得不采取现在流行的异构数据库的方案。通过不断地试错，终于将 POP 平台搭建起来。POP 开放平台奠定了京东的 Java 平台基础，定义了各种编码规划、MVC 框架，为创新地采用 Maven 作为构建工具。这些对京东现在的架构产生了深远的影响。

同年 10 月，思勇开始和邓正平等同事一起开发移动 App。随着公司战略调整，于 2011 年启动电子书项目，艰巨的任务又落在思勇头上。当时完全不懂电子书的思勇和另外一名高级经理李松峰，迅速地组建了一支团队。3 个月便发布了后台管理程序，支持采销录入书籍；5 个月，正式发布第一版电子书客户端。

目前，思勇开始负责网站前台。而前台是京东流量最大的几个板块之一，又

是业务方们的需求重地，技术工作至关重要。

“来京东后，基本上是以加班为主。我个人爱好比较少，以前比较喜欢打游戏，不过现在都不玩了；还喜欢自驾旅游、爬山、看电影、美食。随着年龄增大，歌也听得少了，甚至连以前喜欢玩的轮滑也玩得少了。”思勇如是说。

在平时的接触中，思勇总是给人带来欢乐，幽默的词句不断从他嘴里蹦出。常常很远就能听到他爽朗的笑声。研发的 MM 们都很喜欢他。他无论什么时候都积极地去适应环境，拥抱变化，不断学习和引进新技术，保持乐观的精神去战斗。我想这就是这位“百变神龙”的迷人之处。

## 代码狂——赵刚

拥有完全自主知识产权、世界先进的 Z3D 个性化虚拟试衣技术，虚拟试衣方向牛人

#打造最具影响力的技术团队#



赵刚 16 年来一直没有停止过写代码，写过的代码超过 100 万行，从最初的汇编语言到 C/C++ 语言，再到高级动态语言均涉猎很深。如果代码可以绕地球的话，赵刚写的代码可以绕地球不知多少圈了。他是北京航空航天大学通信与信息系统博士，同时还是 CSDN 发起的 CTO 俱乐部专委会会长，以及 iTechClub 资深会员。加盟京东研发后致力于云计算时代智能化移动端的研究与开发，是京东虚拟试衣项目负责人。

京东移动在 iPad 端的虚拟试衣项目，一直是移动部引以为豪的重点创新项目。赵刚作为核心技术发明人，带领创新团队开发出基于尖端的计算机智能与仿真技术的京东个性化网上试衣间，同时也开发出了可以分发给第三方卖家使用的快速服装素材生成工具，较好地解决了长期困扰服装电商界的用户个性化试衣、低成

本仿真、多层次服装搭配、低用户门槛和高流畅度等难题，其相关的6个发明专利已经通过公司审核并提交至专利局审批。

通过对 iPad 版本上线用户使用数据的分析，赵刚团队对产品的发展路线进行了优化，提出了基于“看→试→做”路径的产品设计思路，并将该思路用于 iPhone 端版本的设计与开发。

虚拟试衣项目的成功运营，后台服装数据制作与管理工具是关键系统之一。只有服装能够快速制作上线，成本控制在商家可以接受的范围内，且技术操作难度较低，才能吸引用户和商家的使用。因此，赵刚主导设计的服装制作与管理后台（CMS 系统）又是一项业界的创举，第一次让商家自己参与服装制作与发布，解决了服装制作慢、成本高的关键技术问题。

京东个性化网上试衣间在业界具有领先性，对提高京东自营和 POP 平台的服装销量具有重要促进作用，给公司创造了巨大价值，同时也对提高京东用户活跃度和流量具有积极作用。

赵刚平时特别喜欢钻研，对新技术非常感兴趣，经常为了优化一个算法在电脑前一坐就是一整天，废寝忘食。他长期研究计算机图形图像与智能技术，拥有多项技术专利与软件著作权。其平时的业余爱好除了与家人爬山就是写代码，16年间一直坚持不让自己离开一线开发。

对于来到京东后的感受，赵刚说道：“京东作为电子商务业界的领导者，其成功不是偶然的。除了商业模式、商务时机等因素外，其良好的技术研发环境、鼓励创新的机制、不惧怕失败的精神，以及公司对技术研发和创新的大力投入、对技术人员重视和关怀，也的确为技术大牛们创造了广阔的发展空间。海量的交易、高效率的仓储和良好的用户体验，需要先进的技术支持，京东已经做得很好，但京东还有很多地方可以做得更好。这里是业界牛人可以发挥专长的地方，京东欢迎每一个有理想有能力的牛人加盟。”

## 青龙之父——李东

物流供应链系统方向牛人

#架构工作是一个不断重构自己思想和追求用户极致体验的过程#



在京东研发，有几个系统不得不提，比如交易系统、POP 系统，还有青龙系统。青龙系统是京东物流核心系统之一，目前每天有 3 万多人同时使用，需要处理所有京东物流配送的订单。这个系统的总架构负责人，就是李东。

2011 年中，青龙团队三十多人开始设计开发青龙系统。经过半年多开发和测试，系统于 2012 年 4 月在华中区上线试运行。2012 年 10 月，青龙系统在全国各区上线正式运行。到了 2014 年 6 月 18 日的店庆日，青龙系统已经可以处理四百多万订单。

2011 年青龙系统进行设计时，公司当时还没有 SAF 支持，李东自己开发了一个简单的框架用于青龙系统。此后，他不断总结和思考青龙的架构，从中提炼出 Clover 分布式任务这一系统。经过他两年多业余时间的开发，目前这个系统支持 Hessian、REStful、Dubbo、Soap 等多协议服务的集成，支持各种协议服务的分布式调度，支持各种协议服务的异步调用和 JMS 协议转化；还能支持多集群部署。Clover 在很多系统上取代了公司以前的 Worker，在易用性、扩展性及稳定性上也比以前增强了很多。目前 Clover 有两个集群、两百多个服务在运行，涉及仓储、物流开放、CRM 等系统。

青龙系统复杂度高、子系统多，现场作业复杂，有正向流程和逆向流程，还要和自动化设备进行对接。如何让这些系统能够协同高效地运转，如何能够让现场人员更高效使用青龙系统，李东及团队在架构上做了很多的创新。

1. 完全采用 SOA 架构，十多个子系统按层进行划分，降低子系统之间的耦合；
2. 其次生产系统和监控查询系统完全分离；

3. 进行了分拣中心缓存和离线生产设计，提高生产的效率；
4. 业务基本采用异步处理，而且以实物为主、信息为辅的流程设计，保证在业务流程复杂的情况下，系统有非常良好的容错能力；
5. 青龙系统服务器比较多，使用了自动化部署和统一日志这些高效的工具，保证上线和排查问题方便；
6. 青龙系统完全采用自动化测试，保证系统的质量。

这些具有前瞻性的设计，实现了当时京东研发部的“多个第一”。青龙系统当时设计的处理能力是每天 200 万单，因为系统有良好的扩展性，目前系统的处理能力远远超过了当时的设计。

生活中的李东是一个爱家、爱 NBA 还爱写代码的好男人，6 岁的女儿今年就要上小学了。对于在京东取得的成绩，他说道：“京东是一个实现梦想的舞台，在这里我成长为一名优秀的架构师，现在正带领着我的架构师团队，去迎接更大的挑战，相信京东，相信团队，相信自己”。

## 纵云者——刘海锋

分布式系统、云平台架构方面牛人

#拥抱变化，迎接挑战，相信技术的力量#



京东（JD.com）自主研发的分布式文件系统、分布式缓存与高速键值（Key-value）存储服务已经成为其开展电商业务的基石。作为中国最大的自营式电商企业，京东用 10 年的时间积累了超过 4740 万活跃用户。2013 年，京东的订单量达到 3.233 亿，成交总额（GMV）超千亿元。如今，京东已经成为全球 PB 级数据管理俱乐部的成员。而要对 PB 级的数据进行有效的存储与管理，稳健、高效的大规模分布式存储体系至关重要。

而掌握这一切的就是我们给云搭上骨头的人——刘海锋。

刘海锋，京东架构委员会主任、云平台首席架构师、系统技术部负责人，获得中国科技大学计算机系学士与硕士学位，曾任职于百度等公司，专注于互联网基础架构技术研发。

从 2013 年开始，海锋就带领团队自主研发了分布式文件系统 JFS。JFS（即 Jingdong FileSystem）已经陆续实现了海量小文件对象存储、块存储、新图片系统等功能，元数据的结构化存储以及与 Hadoop 的集成正处在研发阶段。他还带队研发了分布式缓存与高速 NoSQL 服务，可支撑数百个业务上千台机器规模。海峰的创新精神帮助他在业务上不断突破。有设计新消息队列、搭建新服务框架、研发新图片系统，以及实现全网 CDN 秒级更新等精彩履历为证。在创造辉煌的同时，海锋也面临一些挑战。

“京东每天的库房记录在十亿个数量级，商品图片总共有几十亿张。这些文件基本上都是 KB 级别的，关系型数据库不太擅长处理这些海量小文件，主要的问题是扩容困难，且需要定期删除。而使用开源存储系统也要面临维护工作繁重，需要定制化开发等问题。”刘海锋说。

反复考察后，海峰提出从开源定制开发转向自行研发，希望以此实现系统的灵活可控，并且获得长期、持续的技术收益。目前已经投入使用的 JFS 1.0 具有类似 Paxos 算法的强一致性，以及无单点故障、无内存索引、透明压缩等特性，有三百余个京东的业务应用在其上运行。

基于内存的键值存储方面，京东最早使用的是 Redis。当独立的 Redis 实例积累到数千个，分散系统的管理难题也随之而来。“内存超标、启动慢、难扩展等问题驱动我们在原有 Redis 平台的基础上去创新”，海锋说。目前，京东内存键值存储系统 JimDB 已经能够满足电商业务所需要的精确故障监测、自动故障切换、两级存储、在线纵向扩展和在线横向扩容等需求，该系统的协议、数据类型全面与 Redis 兼容——海峰的前瞻性令人佩服。

海峰一年的努力使得京东云技术产生了质的飞越。目前海锋整体负责存储、中间件、弹性计算等基础云服务。拥抱变化，迎接挑战，相信技术的力量，这就是为京东云搭建骨头的人，刘海锋。

## 灯塔——彭青

研发管理/高并发系统架构/机器学习、自然语言处理  
方向牛人



#京东永远争第一，细节决定成败#

“电商，是既重‘电’也重‘商’的行业，京东作为一家深入零售大量环节的电商公司，需要大量的技术高手，利用技术手段去优化零售业务。通过大数据驱动，打造一个敏捷的供应链体系。‘未来零售’的相貌，就从我们手中开始。”

当满脸笑容和自信的彭青说出这段话时，他所负责的订单生产体系正在以每分钟数千单的速度，将订单生产到仓库系统中去。同时，他带领的市场与比价团队正一刻不停地为京东采销同事们的定价促销，提供数据参考。

来京东前，彭青曾就职于金山网络、暴风等互联网公司，一直从事技术架构方面工作。来京东前上家公司为安博教育集团，主导构建了一整套符合上市规范的研发流程和文档，培养了一批优秀的架构师。京东 Java 代码规范第一版的原型就是他编写的。

从 2012 年起，彭青的团队开始持续对订单履约 workflow 系统进行架构升级。此次工作的主要目标是构建一个集中控制、便于运营，同时能支持高并发、高性能的订单履约流程体系。京东的业务决定了订单履约流程复杂、分支多，单是涉及到需要改造的系统就多达三十多个。为保证 618 店庆活动，留给他们的时间从方案确定到编码完成只有一个月。时间紧任务重，团队所有人都发挥出了 200% 的热情投入该项目中，团队中每个人都在为方案拾遗补缺，项目进行过程中经常可以看到几个人为了一个流程分支在休息区争得面红耳赤，但用不了多久，他们又跑回工位开始紧张的编码工作。这一个月时间里，团队基本每天都加班到深夜，却无人叫苦叫累。有一天加班结束，大家收拾东西准备回家的时候，一位帅哥突然叫住大家，然后一起合了张影，说是回去向媳妇证明自己是在加班。当时，彭青心里对兄弟们愧疚不已，但也为有这么一个优秀、团结、积极主动的团队而感到自豪。

当 2013 年彭青接手比价团队时，整个团队比较迷茫，因为比价系统似乎只有抓取价格后出具报表的用途。业务非常边缘，但事情又很琐碎。伴随着迷茫情绪的是大量的离职和转岗。接手比价团队后，通过与团队，还有业务方的大量沟通，彭青指明了整个团队的方向，确定了两个目标：一是架构升级项目，即通过优化爬虫算法、匹配算法，让整个系统的各项指标，尤其是爬虫的抓全率和匹配准确率、召回率得到极大提升，以保证系统的数据质量，以及大大提升了业务方对团队的信心；二是确立了以“定价”和“选品”为团队产品方向的思路，提出了“商品慧”这一创新性产品。利用历史抓取数据，进行数据挖掘和机器学习，构建价格模型和选品模型，利用外部友商数据为公司采销同事提供合理的定价建议和选品建议。“商品慧”推出后，得到了业务方的赞赏，也为团队未来发展打下良好基础。

作为一名架构师，在团队和业务中需要起到的作用就是灯塔规划团队目标，进行架构升级，构建更加方便快捷的系统，同时还要能带领团队拼命作战。这样的架构师才称得上一个“牛”字！

值得一提的是，彭青的妻子是一位自然语言处理的博士。受她的影响，业余时间彭青也一直在做一些与自然语言处理相关的研究，主要方向是命名实体的识别工作。这真是“每一位成功男人的背后都有一位伟大的女人啊”。

## 强迫症患者——张克房

基础架构管理、网络运维方向牛人

#创新是强迫症的果实#



每项技术创新的成功背后，都有无数的失败经历。工作就是要有“强迫症”，强迫自己做得更好，强迫自己做得更加完美——这是张克房进入京东以来的经验，在他眼里，没有啃不下的硬骨头，只有没有“强迫症”的团队。

克房 2010 年 3 月加入京东后，主要负责京东数据中心网络、自建 CDN 服务、



域名解析服务、负载均衡服务的架构设计和运维管理，以及系统运维和服务资源管理等基础架构相关工作。他是京东早期运维体系创建和发展的直接参与者，奠定了京东运维部现行的多种技术框架和运维模式。

在克房的主导下，京东自建 CDN 服务从 2010 年年初的 3 个节点，扩容到现在分布在全国的几十个节点，带宽从原来可承载几 G 到目前为止可承载几百 G 的流量，规模和架构都发生了翻天覆地的变化。2012 年下半年开始，CDN 架构从原来使用商业硬件负载均衡和 Squid 架构，逐步升级到现在使用的开源 HAProxy 做七层负载和 ATS（Apache Traffic Server）作为缓存软件，这个时候克房已经开始把 CDN 的运维工作逐步授权给建星开展，他自己作为架构师参与到项目中。在这期间思兴对 HAProxy 进行了各种性能优化工作，包括使用网卡多队列提高内核协议栈在多核 CPU 下的性能，在内核态解决了 HAProxy 多进程争抢接收用户带来的锁开销；修改 HAProxy 源码，减少用户态 CPU 使用率；修改 Linux 内核，解决单机 VIP 数量较多情况下，导致协议栈开销大等问题。“这些已经解决的已知问题，还远远不能满足京东的飞速发展需求，我们还需要持续开发，还需要更多的持续优化工作”，克房充满激情地迎接着未来的挑战。

2012 年，商业硬件负载均衡的性能和稳定性，在京东促销时的大流量冲击下暴露出了各种瓶颈和扩展问题。克房开始组织对线上硬件负载均衡进行升级。经过半年左右的二次开发和优化，负载均衡架构和负载软件的性能优化已经成型，采用开源软件 LVS+HAProxy+Nginx 替换商业硬件负载均衡。从 2013 年年底开始对线上硬件负载的服务进行平滑迁移工作，到目前为止，京东 90% 的业务都在这套系统中运行，它平稳地经历了 2013 年的双 11 和 2014 年 618 的 4 倍多流量冲击。“负载均衡是所有业务系统的服务入口，我们还在不断继续完善和优化，未来会承载着京东所有业务系统度过一个又一个的 618 和双 11”，克房充满信心的说道。未卜先知，防患于未然，敢于攀登技术高峰，强迫自己做到最好，这才是京东的牛人！

“京东是个技术飞速发展的平台，进入京东后，个人技术能力成长的速度非常快，以前工作没有踩过的坑，来京东踩上了；没有用过的技术和设备，在京东也都用上了。创新是被逼出来的，也是‘强迫症’的果实，强迫自己找到解决各种问题的办法，强迫自己把工作做到最完美。如果遇不到问题，碰不到瓶颈，每个人对自己的系统都很有信心，从心底也都不会去主动想着创新出有什么更完美更高效的解决方案。而在京东，有着无数的创新点等着京东人去实现，还有无数没有踩到的坑等着京东人去埋。展望未来，让‘强迫症’感染到团队的每个人，兵来将挡水来土掩”，克房如是说。



## 千库之魂——樊建刚

分布式数据库、运维方向牛人

#打造 NO.1 的数据库技术团队#

樊建刚，07 年加入京东，负责整个京东集团的数据库运维及 DBA 团队管理工作。任职至今，他管理的数据库服务器超过 2000 台，在大规模、高并发、海量数据访问、大数据应用方面，积累了丰富的管理经验；同时研发拥有自主知识产权的“蓝海豚”分布式数据库系统，提交了十多项数据库技术专利申请，是京东当之无愧的数据库技术方面牛人。

樊建刚加入京东的 7 年，也是京东数据库不断发展完善的 7 年。

SQL Server 是樊建刚最开始参与管理的数据库，同时也是京东历史上使用最早、最广泛的数据库，其中使用复制技术的服务器超过一百多台，高峰时一天的复制命令超过 80 亿条，在微软全球客户应用的 OLTP 系统中名列前三。建刚带领的 DBA 团队在复制技术上进行多项创新，有些技术已经充实到微软全球资料库。勇于创新、不断钻研的 DBA 将 SQL Server 的各项高级技术，如群集、日志传送、镜像、复制到最新的 Always On 和内存数据库等，融入到京东的业务系统中，取得良好的效果。至今，其依旧稳定运行在京东的很多核心业务上，支撑从商品到订单再到库存和配送的各个业务环节。

伴随京东整体技术架构的升级，数据库逐步迁移到 MySQL 上，拆库拆表大量应用，给研发的开发效率、DBA 的运维管理带来了极大挑战。面对挑战，樊建刚带领数据库团队研发了拥有自主知识产权的分布式数据库系统——蓝海豚。蓝海豚能将数据库系统自动拆分、水平扩展，对应用程序透明，支持读写分离、负载均衡，这些新特性对提升系统性能起到重要作用。蓝海豚于 2013 年 3 月 28 日正式发布，并用于订单轨迹系统；2014 年 4 月 8 日正式发布蓝海豚二期，已有部分重要系统使用，产生了十多项专利。

蓝海豚的研发成功与 SQL Server 高级技术的成功使用,大幅提升了京东数据库的性能,提高了开发效率,极大降低了数据库运维成本,这也标志着京东的技术水平在业内达到一个全新的高度。

对于取得如此好的成绩,樊建刚相信京东著名的“倒三角”理论:无论是品牌、关键 KPI、供应链还是用户体验,最终都建立在“团队”这一基础上。

“最优厚的工资给最优秀的人才,最优秀的人才创造最大的价值。欢迎所有优秀的人才加入京东,你们将会获得更大的提升空间、更多的施展空间,以及为五亿网民服务的成就感。来吧,加入到我们当中,我们彼此尊重、坦诚相待、虚心接受意见,携手打造 NO.1 的数据库技术团队,共创美好的明天。”

## 数据达人——刘思喆

推荐系统算法优化、大数据方向牛人

#满怀激情,才能积极向上#



2005 年毕业于中国人民大学统计学院的他,先后就职于亚信联创 BOC、神州数码思特奇 DSS,主要为电信运营商提供数据挖掘及业务咨询等顾问服务。

几经辗转,当 2012 年他满怀信心地踏入京东大门的时候,还是被京东的数据惊到。“实在太庞大、太丰富了!”也只有在互联网才可以这么轻松地获取这么详细的各类数据,他不由地被这数据的海洋深深吸引。

“那时候还在国家会议中心附近办公,我清楚地记得,那天下班乘上八号地铁线,想到第二天可以用到这么多的数据做各种 Model,不禁一个人傻傻地笑了。”对于数据达人的他而言,京东无疑是数据科学的潮头,是数据科学家的天堂。当然,这么好的条件,让他更加充满了奋斗的动力。

伴随着京东推荐系统从 1.0 变迁到 2.0，时间到了 2013 年年底。新升级的推荐系统架构已经基本稳定，但同预期的推荐系统能够为公司带来的价值还有一定差异。如何才能提高推荐系统的效能，为京东高速发展保驾护航，是他当时考虑最多的问题。经过缜密的思考后，他采用了以下方法提升推荐效果。

1. 以数据分析为工具，增强对数据和业务的理解；
2. 使用用户反馈数据为依据，融合不同类型、不同维度数据源；
3. 测试不同算法（甚至参数）在不同数据源的效果。

为了验证以上方法，逐步调整至最优策略，他带领团队累计实施上百次实验，几乎所有的推荐位都被实验覆盖。在优化过程中，点击率、点击转化率、覆盖率等指标均有明显提升，达到了预期效果。

有人说：当你将信心放在自己身上时，你将永远充满力量。充满力量的他，一次又一次地缔造着辉煌。

作为推荐团队的主要算法负责人之一，他在半年时间里，带领团队将 PC 端推荐带来订单行占全站订单行的比例，提升近 100%，推荐系统千次请求的销售金额提升 67%。

他，带领推荐离线算法团队将移动端推荐订单占比，由 2 月的 0.5% 左右提高到 6% 以上。

因为热爱，工作之余，他发起数据科学沙龙，参与举办中国 R 语言会议。

他就是刘思喆，一个真真正正的数据达人。

## 思考者——隋剑峰

网站交易、架构方向牛人



#拥抱变化！京东的业务、技术和管理都在快速发展和变化，需要我们主动去迎接变化，而不是等待变化的到来。#

“生活上有车有房并给小孩落户西城，上学问题也解决了，所以生活上完全没有压力，工作上就可以完全按照兴趣来做事情，非常满意当前的工作状态。”当剑峰说出这句话时，脸上带着满满的笑意。

隋剑峰，京东架构师，主要负责交易平台的相关架构和核心技术研究、升级工作，包括京东主站主页、商品页、列表页、频道页等主要入口页面。在入职京东前，他在新浪和搜狐呆过五六年，一直从事互联网技术工作，拥有非常丰富的行业经验。

拥有丰富行业经验的他，在京东这个更广阔的舞台上充分展现了自己的才华。其中最让他自豪的一个项目便是实时价格实现项目，该项目实现了京东商品价格的秒级更新显示。

最初，京东的价格还是采用图片显示，并且使用了 CDN 缓存，这存在一些严重问题。

1. 价格展示不实时，存在 5~10 分钟的延迟（如果放在现在，可能会因为价格欺诈天天被投诉）；
2. 资源占用严重；
3. 价格更新频度高，导致 CDN 的穿透率很高，命中率就相应地非常低；
4. 一个页面存在很多的商品价格，每个价格都要占用一个单独请求，导致请求资源量很大。

面对问题，隋剑峰冷静分析，将价格图片显示改为实时价格实现，并采用 Nginx+Lua+Redis 模式，服务器从上百台节省到 10 台的规模，实时性达到秒级实时，并且支持多个价格合并请求，大大降低请求资源量。

另外一个重大项目是预售秒杀从主交易流程独立。该项目大大降低了热点商品被经销商和用户恶意下单的数量，以及主交易流程被机器人刷新造成的系统开销和风险。

当时，随着华为荣耀 3C 产品的热度增加，导致很多经销商无论有货无货都不停通过刷新京东的交易系统来抢购产品。尤其在促销的时候，交易系统的访问量达到同期的 100~1000 倍，导致系统压力非常大。因为经销商使用机器人抢购，导致每次活动的正常用户基本不可能抢购成功。做活动的初衷是让利给用户，结果被经销商抢到的再加价卖给最终用户，用户最终未能受惠。

面对严峻的现状，考虑到交易系统本身的复杂性导致无法很好解决这个问题，隋剑峰创新性地将此类特殊的业务，从主交易流程剥离出来，并考虑深远，促使未来所有促销模式均接入新的预售秒杀流程。

从效果看，预售秒杀从主交易流程独立，很好地控制了经销商下单所占比重从原来的 99% 下降到 10%；通过改变部分交易流程，很好地控制了机器人刷新系统的频率，保护了系统，这样就使系统可以有更多的资源，用来做更复杂的风控逻辑，更加准确地识别用户。卓越的成效，使得该项目获得 2014 年第二季度技术创新第一名。

务实、创新、懂得合作、擅于思考总结的隋剑峰，将京东交易的架构水平带到了一个全新的高度。隋剑峰不仅仅技术一流，更乐于分享他辛苦工作收获的每一点经验。

他常说，要结合业务做技术，把核心的技术用在核心业务上。在行动中思考。边实践边思考。比如，有的问题不能一下子都解决好。都想清楚，可以采用切流量上线，上线前数据比对以及线上数据镜像压测等方法基本能保证系统的可靠性。还有就是技术先行，很多系统总是存在很多的历史问题，技术上要在系统出现瓶颈和问题前就给出解决方案，等待时机将新技术应用到线上业务系统。而这需要至少三个月或者半年的技术储备。

“边实践边思考”，这是到京东后很多技术大牛们共同的感受，因为京东的发展变化太快，很多时候技术是在满足业务的需求。正因如此，我们更需要像剑峰一样的思考者和实践者。只有不断试错，才能趟出一条路来。

## 问题终结者——王春明

数据处理、站内搜索方向牛人

#做执行力最强的团队#



“京东技术团队可能不是业内研发技术最强的团队，但绝对是执行力最强的团队。”王春明的话，透着他他对团队的骄傲，对京东未来的信心。

2011年8月刚加入京东时，王春明是站内搜索开发经理，负责站内商品搜索、列表页后台和各垂直搜索后台。从此开始，他组织并参与过很多次搜索架构升级，但最让他记忆深刻的是第一次升级的项目。那是他进入京东后做的第一个大项目。

当时，搜索采用的是 Lucene，遇到很多用户体验的问题，无法解决。这个项目目标就是，自行研发一个搜索引擎代替 Lucene。遇到的主要挑战是，切换既要保证系统的稳定，还要支持系统原有的业务和新版开发过程中老系统增加的业务，更要保证新系统的用户体验要优于老系统。而整个搜索团队不到 10 人，其中还有 1/3 是实习生。艰难的处境挡不住迎接挑战的热情，王春明迅速组织人员进行任务分拆，每个人都负责一大块内容，发现问题就在工位旁把大家拉到一起讨论。讨论没有结果的，就去技术论坛找各种解决方案不断尝试。新版遇到很多的性能问题和内存占用超高问题，都是这样解决的。后来，新版排序遇到 Bad Case 时，他们一次次调整参数，找出最新模型，然后进行回归测试。最终使用新架构代替了 Lucene，应用于主搜、列表页等众多场景。

这次升级，提升了系统稳定性和可扩展性。满足业务需求的同时，解决了日益增长的数据量和访问量给搜索系统带来的压力问题。其间系统承受了多次 618、双 11 的考验，运行稳定。

实时索引项目，也同样让王春明记忆深刻。由于京东平台的商品量快速增长，之前基于时间戳增量的功能实现起来越来越慢，无法满足业务需求。于是，王春明带领团队开始做实时索引项目。这个项目将增量索引修改为基于消息实现，项目调整的困难主要在于实时索引会降低系统的性能。在提升索引实时性的同时，

必须要减少其对性能的影响。为此，王春明对索引结构和查询流程进行了优化，最终系统上线之后商品在搜索页的展示延迟控制在 10s 之内，成功解决了之前延迟过久的问题，并且成功降低了性能损耗，线上服务器成本没有明显上升。

三年前，刚刚从芯片设计公司转行互联网进入京东的王春明，不会想到他会在一个对他而言全新的行业和公司取得这样大的成就。现在的他是站内搜索的负责人，负责的重要应用有站内搜索、移动端搜索后台、微信商品搜索后台、Jshop 店铺搜索/列表后台、PC/移动列表页后台。这些页面每日访问量总计超过 1 亿次，带来的销售额占京东总销售额的比例超过 70%。

“进入京东后，京东平台的快速扩张，给我们研发提出了一个接一个问题。这些问题很多都是京东特有的，业内基本没有好的解决方案，对技术的挑战很大。解决这一个个问题的过程很痛苦，有很多时候我都觉得扛不过去了。但是我们的团队给了我坚定的支持，最终解决了这一个个问题。”正是这些经历，让王春明变得越来越牛，“京东平台仍在快速扩张，挑战性的问题较之前有增无减。期待更多的牛人加入团队，有问题我们一起扛。我们一起在解决问题的过程中变得越来越牛。”他翘首以待更多的挑战。

## 四驾马车的车轮——陈成钢

基础运维方向牛人

#人在很多时候，缺少的不是做事的方法，而是一往无前的决心和魄力 #



2013 年，京东公布了未来四架马车的宏伟战略。对于京东运维部的创始人之一陈成钢来说，这意味着他和他所在运维团队——这辆四架马车的车轮，必须能抵抗四匹骏马飞速奔跑带来的压力。这是个极大的挑战，核心网络、CDN 系统、DNS 系统、负载均衡系统、基础设施建设的每一个环节都是保障网站及各项业务



系统稳定可用的关键。

对此，陈成钢这位在北京工作了 14 年的东北汉子，只是淡淡地说：“我在经历了惶恐、忐忑、淡然的心路历程后，已能做到运筹帷幄。”

陈成钢于 2007 年作为开发人员加入京东。2008 年京东开始创建运维部，将应用运维及网络环境进行单独管理。于是他转行做运维，作为创始人之一开始搭建运维部。

但当时，运维只具备服务器管理及网络管理的能力。2008 年京东网站进行了从技术架构到前端展示的全面改版，运维的难度也直线提升了几个等级。成钢一个人应付着几乎所有的系统上线以及故障处理。凭着 7 年的研发经验，对于各个系统的问题及故障，他还都能快速定位并迅速解决。但当时京东所有的系统均采用 .Net+SQL Server 技术，维护复杂，批量维护更是不可能。疯狂的时候成钢甚至直接修改开发人员的代码和 SQL 语句以解决线上故障。对成钢来说，2008 年是极其痛苦并快速成长的一年。

京东网站改版后，必须进行 CDN 的规划部署。开始成钢并不想自建，想使用第三方的服务，毕竟自身的技术实力还不够。但服务商的一次故障彻底改变了他的想法——CDN 服务商的机房事故，导致了网站无法访问长达一个小时之久。从那时起，自建 CDN 这个想法就无法从成钢脑海中抹去了。

2009 年年初，京东的第一个自建 CDN 节点在天津诞生。这第一步走得很是艰辛，技术选型、网络规划、软件的开发、测试、部署、上线、排查故障，全靠自己一点一滴的积累完成。从这个 CDN 节点开始，京东从无到有、逐步完善。到今天，CDN 节点已遍布全国。单个节点承载能力从开始的 1~2G，已升级改造成万兆节点，可承载 20G 以上带宽容量。首屏时间从当年的 3 秒多到现在平均 1 秒，全国的客户均可以按照自己所在地域就近访问，全面提升了用户的访问体验。京东建成了具有自己特色的、独一无二的 CDN 系统，而这正是成钢和他的团队一手创造出来的！

也许你无法想象，一个年销售额达到亿元的网站，其运行的生产环境网络的核心设备竟然是一个百兆的 Tplink 交换机。可事实就是如此，为了换掉这颗埋在心脏上的雷，以及避免未来还有这样的雷出现，一个长远的网络规划是必需的。京东的业务增长每年以超过 200% 的速度在飞奔，从 2009 年开始，服务器、交换机、机柜数量都大幅度增加。刚刚排除掉百兆核心那颗雷，网络发展又跟不上公司的速度了。公司业务跑得太快，相应带来的系统压力持续上升，唯一的解决办法就是改造。虽然困难重重、荆棘满路，但成钢一直坚持，坚信没有什么问题是解决

不了的。扩带宽、改架构、割接网络、迁移应用系统，以及一轮轮的设备迁移、系统迁移，为的就是平滑割接，不中断数据传输，不影响前端客户访问。几年来，数据中心承载能力从只能接入 200 台以内服务器、百兆核心，过渡到千兆网络、可容纳几千台服务器，再到目前的万兆网络环境，服务器可容纳量和网络吞吐量已经实现了几千倍的扩容。从“定时抢购”到“秒杀”，再到“8.15”价格战，以及每年的“618”、“双 11”，经历了一个又一个的大战役。每个大型促销活动，都是对成钢这些京东运维人的考验。

“坚持，是这些年我做运维的信念。运维的工作就是持续优化，把一件事情不停地做，不断地完善。”这位东北大汉骨子里就有着豪爽、顽强、坚韧的性格，也正是这样，才能担起四架马车车轮的重担。正如一句话所说：一辆马车能跑多快多远，不仅取决于马的速度，还有车轮的质量。

正是因为有无数和成钢一样的运维人，为京东这架马车保驾护航，京东才得以飞速向前。

## 知心爱人——刘尚堃

数据搜索推荐方向牛人

#这里不怕犯错，也给你机会试错，只怕你自己赶不上平台发展的速度 #



2011 年加入京东伊始，尚堃就开始负责京东的搜索推荐方面的研发，担任京东搜索引擎主管。他带领团队自主研发的搜索引擎于 2012 年年初上线。2012 年下半年，尚堃主导进行京东广告算法的优化，是京东广告系统的奠基人，广告平台于 2013 年上线。2014 年起，尚堃正式全面接手数据部搜索推荐架构研发、算法研发、产品运营相关的工作。

3 年过去，他凭自己的努力，从主管升到副总监，其所带领的搜索推荐团队也从 3 年前的小猫两三只，到现在 70 人的大团队。他及团队在京东大数据的挖掘和应用上做出了不可磨灭的贡献。

2011 年京东的搜索使用的是开源的系统，在他的带领下，2012 年年初京东上线了采用 C/C++ 研发的自主搜索引擎 Jsearch 1.0，一方面性能大幅度提升，单台机器 QPS 在 200 以上，另一方面工程师在上面开发算法可以得心应手。当年一系列算法上线之后搜索引擎的转化率翻倍。在这之后又研发了 Jsearch 2.0，目前支持京东 Web 端、移动平台、微信/手 Q 平台搜索和类目的服务，日总搜索量过亿，按照搜索量计算京东的搜索引擎已经是国内第五大搜索引擎。在这之后，2014 年中尚堃带领的团队再接再厉，上线了 jsearch 3.0，实现了可以承载海量 SKU 的分布式搜索引擎，并且做到了排序和架构分离，研发出了坚如磐石的搜索引擎，同时可以让业务在上面更好更快的发展。

2012 年尚堃开始负责广告算法团队，成为京东快车广告系统的奠基人。2013 年年初广告系统上线，为公司带来大量纯收入。在这过程中他主导研发的 CTR 预测算法和广告排序算法，对提升点击率和收入起到至关重要的作用。

京东的推荐引擎在他们的努力下，RPM（每千次请求产生 GMV）翻倍。推荐产生的 GMV 提升了 300%。同时，他们还完成了分布式搜索的开发。目前，搜索和推荐在 PC 和移动端的订单行占比达到了 50% 左右，居行业领先地位。

从完全依靠外部搜索到一步步构建自己的推荐搜索引擎和平台，可以说以尚堃为代表的京东人，从无到有淌出了一条属于自己的道路。“加入京东其实就是与这个平台一起快速的发展。”回忆过去，尚堃觉得最大的收获就是看到团队在自己的带领下飞速成长。比如，很多以前不是做推荐搜索的新同事，在半年或一年后就成为了该领域的牛人。

## 618 之盾——王晓钟

交易系统架构方面牛人

#没有技术做不到的事#



说到王晓钟，就不得不提每年电商的两次大促——618 和双 11。这两次大促与他在京东的技术大牛之路，结下了不解之缘。

作为研发人，最怕也是最期待的，莫过于这两次大促期间带来的高流量、高并发对系统的挑战。晓钟加入京东不到 1 个月，便参加了当年双 11 备战，然而这第一次的经历不太美好。

当时系统架构落后，双 11 当天不断宕机，技术高层被请去喝咖啡，更有加一倍服务器的传闻。作为一名交易系统的架构师，晓钟觉得这是奇耻大辱，发誓要打翻身仗。然而当时的交易系统面临着两个难题：一是正在从 C# 升级到 Java 的关键时期，所有系统都是 Java 和 C# 版本并存；二是服务化做得不够彻底，多个关键系统无法水平扩展。

幸运的是整个交易团队斗志昂扬。他们在技术上首先建立了完备的交易系统监控，包括软件和硬件，明确了现有系统的瓶颈；其次，务实地分析 C# 版本交易系统的瓶颈，不同情况不同对待，关键系统全部设计成支持水平扩展；最后，和交易的兄弟们一起努力实现架构规划，一夜夜的通宵成就了第一代 Java 版的交易系统。第二年的 618，交易系统表现稳定，让晓钟和他的团队长舒了一口气，当年双 11 时也平稳度过，再次证明了系统架构的合理性。

从此，晓钟便成为了 618 和双 11 的方面牛人，带领交易系统团队不断升级，不断迎接下一次更大的挑战。

今年的 618 也是上市后的第一个 618，大家都压力山大。值得庆祝的是，交易系统一切正常，运行平稳。可以说，京东网站前端最重要的就是交易系统，只要有了稳定的交易系统，大家就可以把心放到肚子里。

对于京东，晓钟如此说道：“如果你喜欢挑战，请加入京东，这里有极好的

舞台任你发挥。大流量，高并发，高稳定，不管是基础架构还是系统架构，都需要优秀的人才去设计。如果你务实，请加入京东，这里有大量实际系统问题等你解决。在规定时间内完成不可能的任务带来的成就感不是哪里都能找到的。加入京东吧，你不是一个人在战斗，和有着相同价值观的神一样的队友一起战斗是人生的极大乐趣。”

## 创新先驱——杨凯

电商领域业务架构方向牛人

#我们都是平凡人，我们唯有靠智慧和汗水，靠比别人多流一滴汗，才能取得成功#



杨凯于2010年3月来到京东研发部，来京东之前一直从事互联网、电子商务领域的技术工作。在京东工作期间做过产品、写过代码、管过项目、带过团队，主导过数十个成功的项目，在电商业务和技术方面有着比较丰富的经验。2011年5月通过层层PK获得了公司第一届管理干部轮岗培训的机会，1个多月的各业务岗位工作使他学习到了公司的整体业务，一线工作让他更加容易理解用户的需求和困难，也正是这次机会帮他找到了自己非常喜欢的工作方向——电商领域业务架构。

他来京东负责的第一个项目是POP开放平台商家和商品系统建设。最终设计出业界领先的商品和类目数据结构，充分考虑了商家海量的商品和灵活多变的品类结构，第一次把商家和店铺的概念引入京东的系统，很好地支撑了初期商家入驻和丰富商品这两项重要工作，为POP业务整体的推进做好了铺垫。

随着POP平台商家和商品系统逐渐成熟，市场出现了一个新的宠儿——团购。2010年11月下旬接到任务，需要在同年12月15日前完成团购系统的研发上线。当时包括他在内只有3名研发人员。他非常清楚，在这个时候公司更追求业务尽早开展，而并非是花大量时间做出一个完美的系统。通过分析，选择购买了一个

PHP 版本的团购系统进行改造，和京东主站实现了用户、商品、支付、订单等系统对接，重新实现团购发码和售后退款业务。历时 3 周的研发测试，团购系统以惊人的速度于 2010 年 12 月 14 日正式上线运营，为京东在团购市场占领一席之地争取了时间。

杨凯总是对创新产品充满热情。2011 年年底，他开始主导 JSHOP（JSHOP 系统是公司 PC 端、移动端活动专题和商家店铺的装修和展示平台）整条产品线。产品主要包含活动专题和商家店铺两部分。活动专题系统成为发布促销活动、专题购物、618 与双 11 主会场的唯一入口；商家店铺成为京东所有商家的门面，也是商家获得订单的主要入口。在公司移动化的战略下，今年移动端的活动和店铺平台也投入了运营。他坚信 JSHOP 产品未来会以更广的服务范围，更多的产品创新为用户、商家、京东持续不断地创造价值。

“在京东的这几年，我觉得自己在技术和管理的方方面面都有了很大的进步。但最大的收获还是在企业价值观和文化培养下，我成为一个信守承诺，追求合作共赢，懂得站在用户角度思考问题的人。有了这个基础我才有机会走上核心工作岗位去提升自己，才能做好各个岗位的工作。京东是一个可以承载梦想、高速发展的舞台，在这里有无数的机会等你来抓住并接受挑战，在这里你会看技术创造的价值是多么的伟大！”杨凯如是说。（作者/刘家瑞）

我曾经在很多场合谈到京东可以高速发展到今天的规模的原因，其中最核心的是坚持“倒三角”战略：建立出色的团队；打造财务、物流和技术三大核心系统；降低成本、提升效率；为用户带来最佳体验。在《京东技术解密》一书中，大家会看到技术驱动的力量，这也是京东第一次全面展示自己的技术体系。

——京东集团创始人兼首席执行官 刘强东

作为一个消费者，我很喜欢京东网购正品和快速这两大特点。京东仅用十年便快速崛起，成为千万消费者喜爱的电商品牌。要将一个大型服务平台做到优质已非易事，做一个可信任可依赖、服务范围覆盖全国的电商生态，更是无比艰难。通过《京东技术解密》一书，读者可以感受到京东对技术架构永无止境的追求。

张志东

一流的互联网企业离不开一流技术，京东能成为全球排名前10的互联网公司，技术是其高速发展的关键因素。很多传统企业进入电商领域，没有认识到技术管理、研发的不同，吃了极大苦头。而京东技术部门用10年时间，从30人发展到4000人，营业额也成长上千倍，背后的技术如何发展，又碰到哪些挑战，本书一一做出全面而详细的披露，因而其对于中国电商行业及互联网行业都是极其宝贵的资料。

——CSDN和《程序员》创始人 蒋涛

京东商业上风生水起的背后，技术的力量不容小觑。商业的竞争，说到底还是人才的竞争；互联网商业的竞争，说到底比拼的是你的技术积累厚不厚。听说“京东技院”已经开张，相信会对京东技术的下一个10年起到积极的作用。

——InfoQ中文站联合创始人兼CEO 霍泰稳

一个技术团队，用满满的自豪感讲自己的故事，然后就有了这本书。那些影响重大的项目，那些“高大上”的技术挑战，还有那些小八卦，作为技术人，一边读一遍YY，很过瘾。这本书内容非常全面，也刷新了我对一个大型电商系统的复杂性的认识，京东人的自豪感，是有理由的。把京东作为首选线上购物平台的用户，对京东又多了一分信心。

——豆瓣技术副总裁 耿新跃

这不是一本传统意义上的技术书，而是讲述了京东技术和成长的故事。10年时间，京东在技术团队的支撑下，从一个简陋的网站和混乱的库房，迅速崛起为国内一流电商平台。《京东技术解密》揭示了技术和产品人员对京东崛起的重要作用，涵盖大型电子商务系统发展过程中所有的技术演进和产品创新点，包括编程语言、技术架构、大数据、交易处理、产品创新与融合、运维运营等。愿意了解互联网和电子商务体系的从业人员，都应该读读。同时也希望京东技术团队坚持开放、拥抱开源，早日成为“中国最具影响力的技术团队”。

——MacTalk出品人 池建强



博文视点Broadview



新浪微博  
weibo.com

@博文视点Broadview



策划编辑：张春雨  
责任编辑：徐津平  
封面设计：吴海燕

上架建议：电商网站/技术人文

ISBN 978-7-121-24691-3



定价：65.00元