TiD2016
质量竞争力大会
软件研发顶级盛会

下一代 SOFTWARE DEVELOPMENT 软件研发
更高 更新 更深

重构

如何在互联网+的时代应对技术转型

范钢

关于我自己

TiD2016

范 钢

- 航天信息高级架构师
- 《大话重构》作者
- 架构设计与重构客座讲师
- 大型遗留系统改造专业户
- 长期关注大型业务系统的系统优化、防止腐化以及技术改造等困扰软件企业的问题
- 帮助大家进行互联网转型

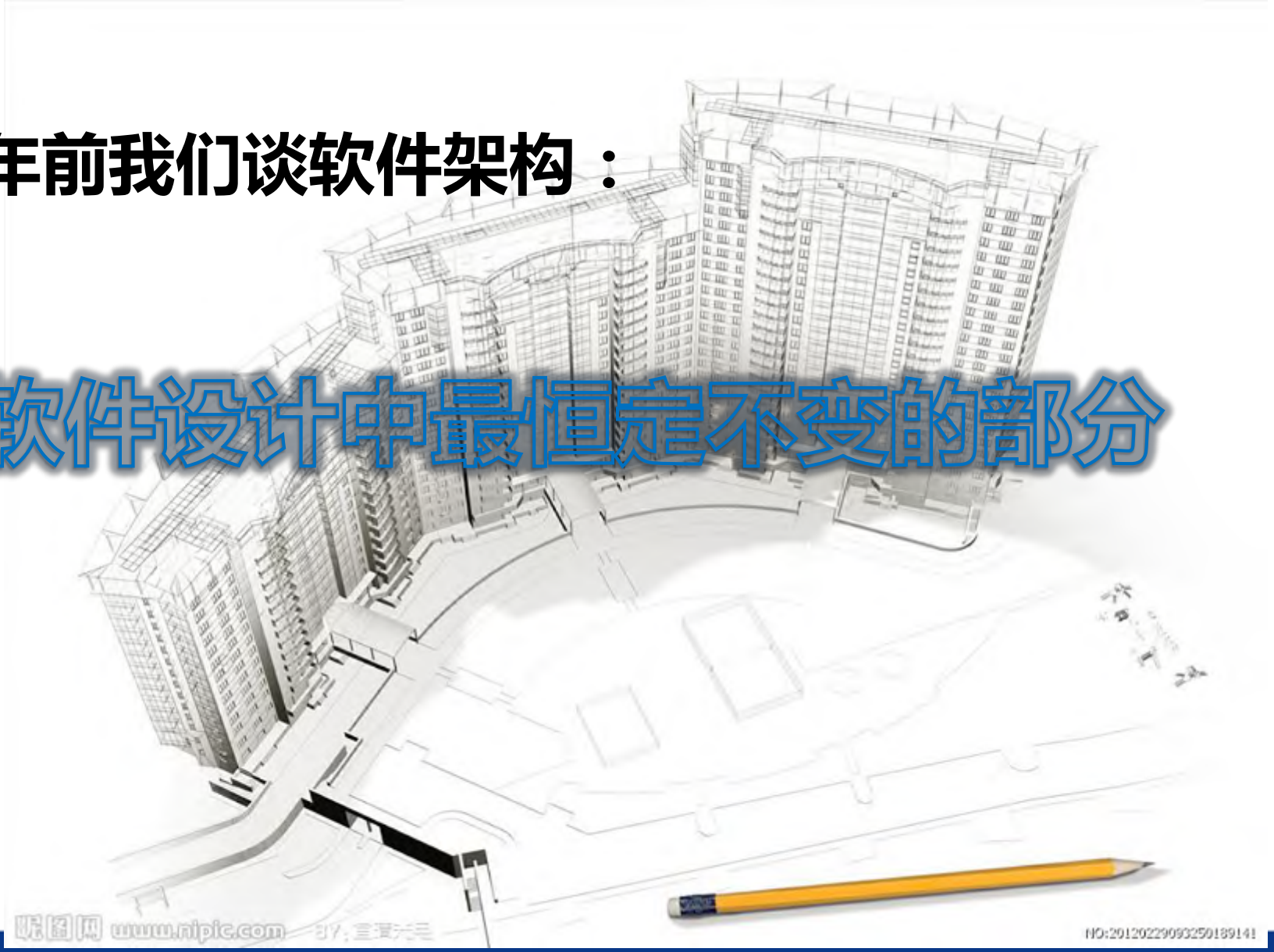
我在Tid大会上的演讲

TiD2016

演讲内容

Contents

十年前我们谈软件架构：



我们的软件设计正在发生着翻天覆地的变化

TiD2016

资讯

业界 移动开发 云计算 软件开发 程序员 极客头条 专题

大数据 数据中心 服务器 存储 虚拟化 NoSQL 安全 云

今天我们谈软件架构：

CSDN首页 > 云计算

58同城沈剑：好的架构源于不停地衍变，而非设计

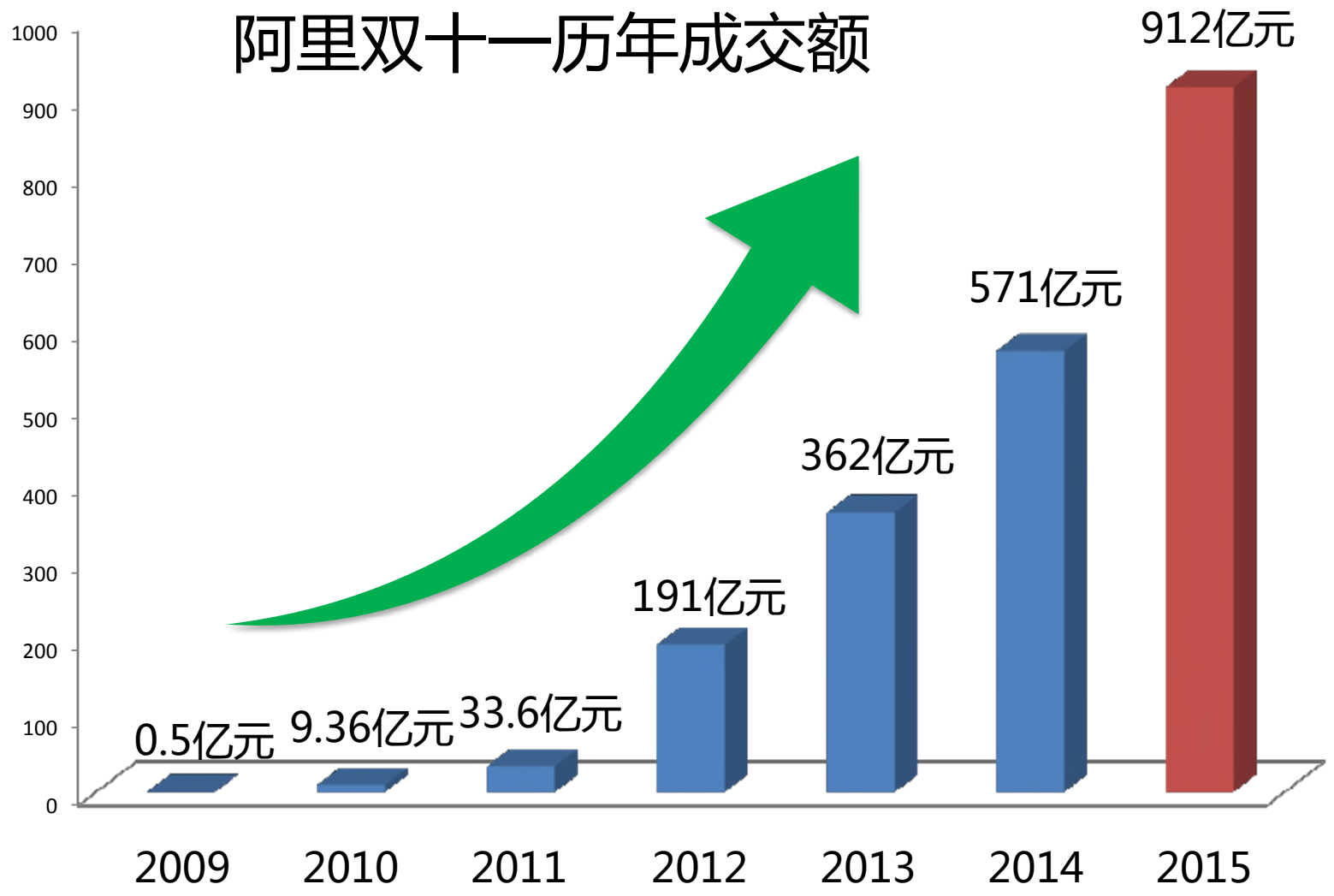
发表于 2015-10-24 01:37 | 7348次阅读 | 来源 CSDN | 55 条评论 | 作者 沈剑

大数据 58同城 架构

摘要：对很多创业公司而言，很难在初期就预估到流量十倍、百倍以及千倍以后网站架构会是什么样的一个状况。同时，如果系统初期就设计一个千万级并发的流量架构，很难有公司可以支撑这个成本。

【编者按】对很多创业公司而言，随着业务增长，网站的流量也会经历不同的阶段。从十万流量到一百万流量，再从一百万流量跨越到一千万甚至上亿的流量，网站的架构需要经历哪些变化？在“OneAPM 技术公开课”第一期中，**58同城的技术委员会执行主席沈剑**对此进行了详细剖析。

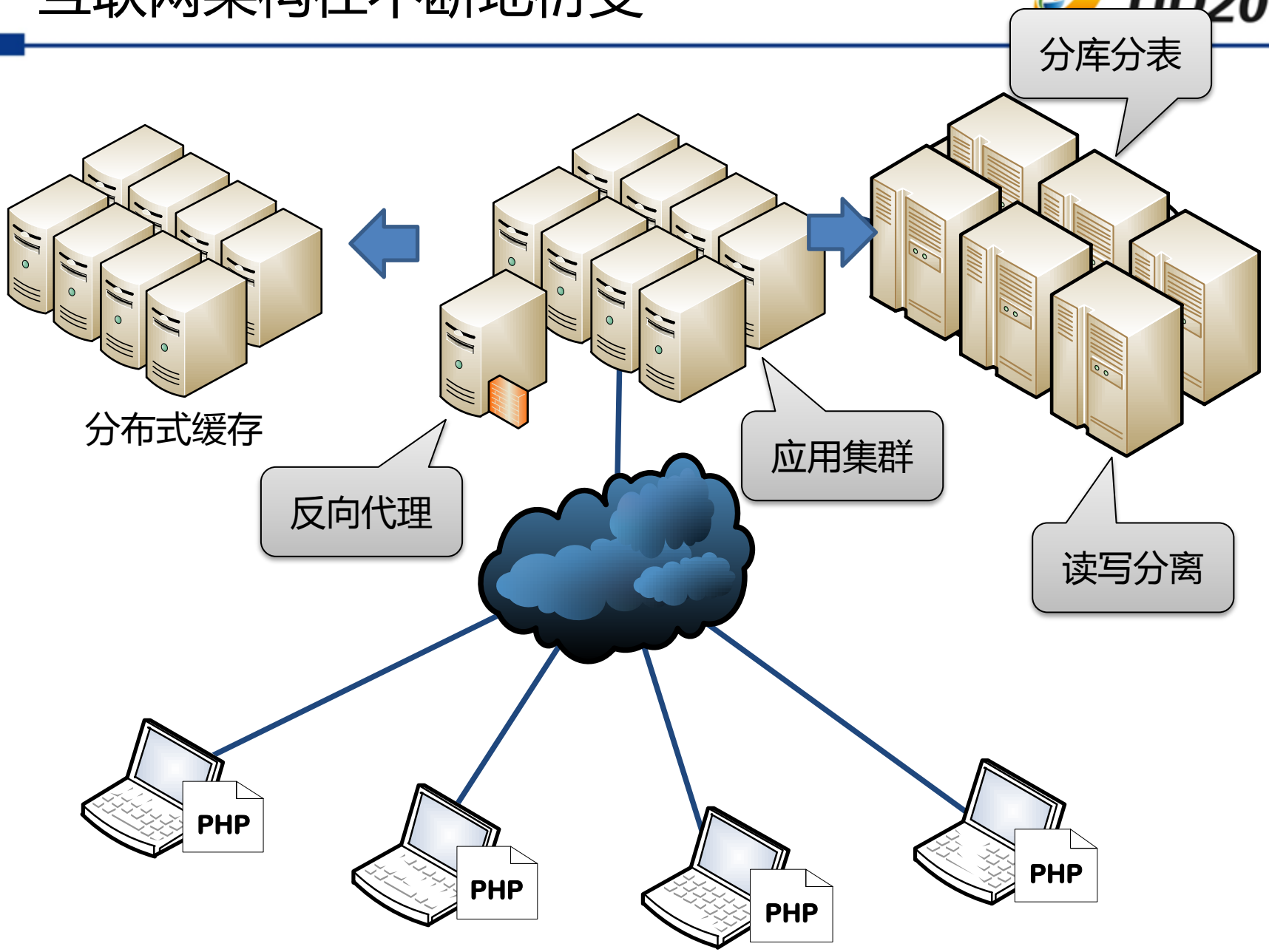
互联网应用推动着架构的衍变



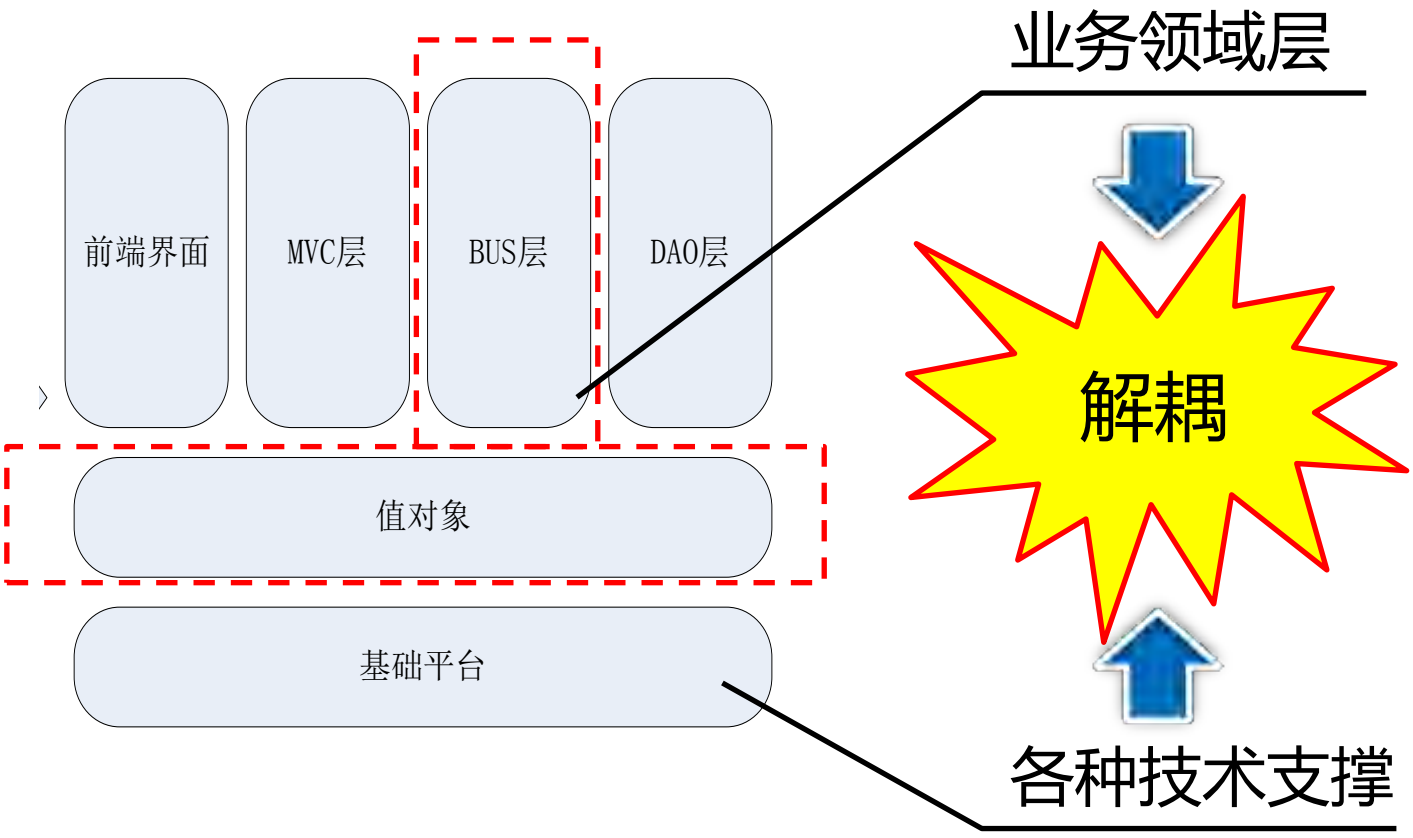
互联网转型推动着传统行业的改造

- 互联网+春运=12306
- 互联网+打车=滴滴打车
- 互联网+金融=P2P
- 互联网+教育=在线教育
- 互联网+税务=电子税务局

互联网架构在不断地衍变



以往的架构设计已不适应架构的衍变



渐进式技术改造过程

解决技术改造的困局

系统必须要进行技术改造

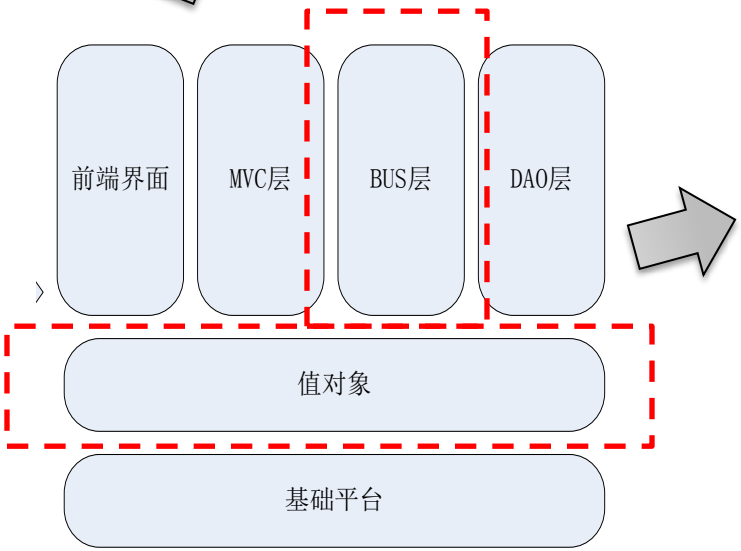
技术改造**难**

渐进式技术改造过程

1. 让系统适于技术改造
2. 进行相应技术改造

平台建设与系统重构并行开展

代码优化



平台建设

TiD2016
质量竞争力大会
软件研发顶级盛会

下一代 SOFTWARE DEVELOPMENT 软件研发
更高 更新 更深

互联网不再是推动系统改造的

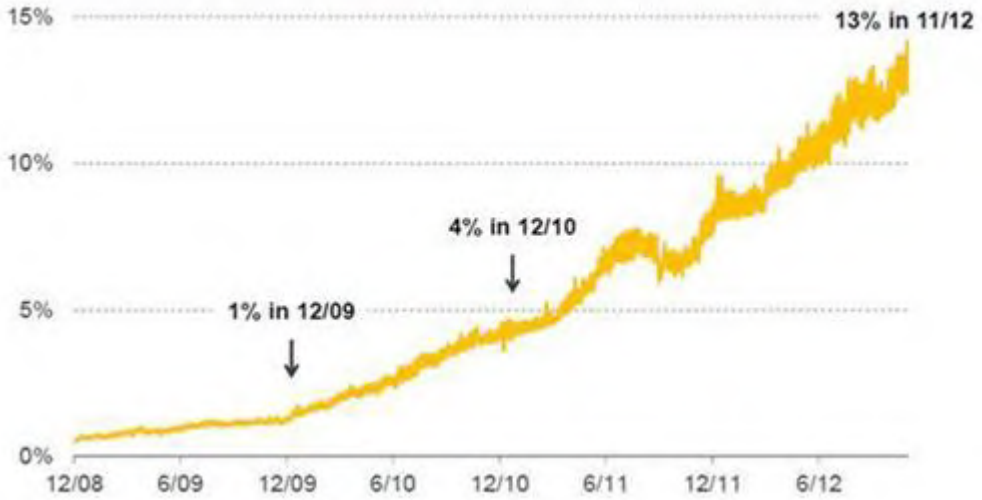
唯一 力量

大数据即将引领整个产业的

第二次改造

互联网应用引爆数据量的迅猛增长

2009-2012 移动访问量占全球互联网访问量百分比（半年度记）



2010-2020 全球大数据规模增长图例。单位EB（1EB=10亿GB）

传统数据库已不堪重负，NoSQL数据库引领变革

传统关系型数据库

扩展困难：由于存在Join这样的多表查询机制，使得数据库在扩展方面很艰难

读写慢：当数据量达到一定规模时于关系型数据库的系统逻辑非常复使得其非常容易发生死锁等的并发，所以导致其读写速度下滑非常严

成本高：企业级数据库的License很惊人，并且随着系统的规模不断

有限的支撑容量：当磁盘容量已经满足数据量需求时，只能整体数据

NoSQL数据库

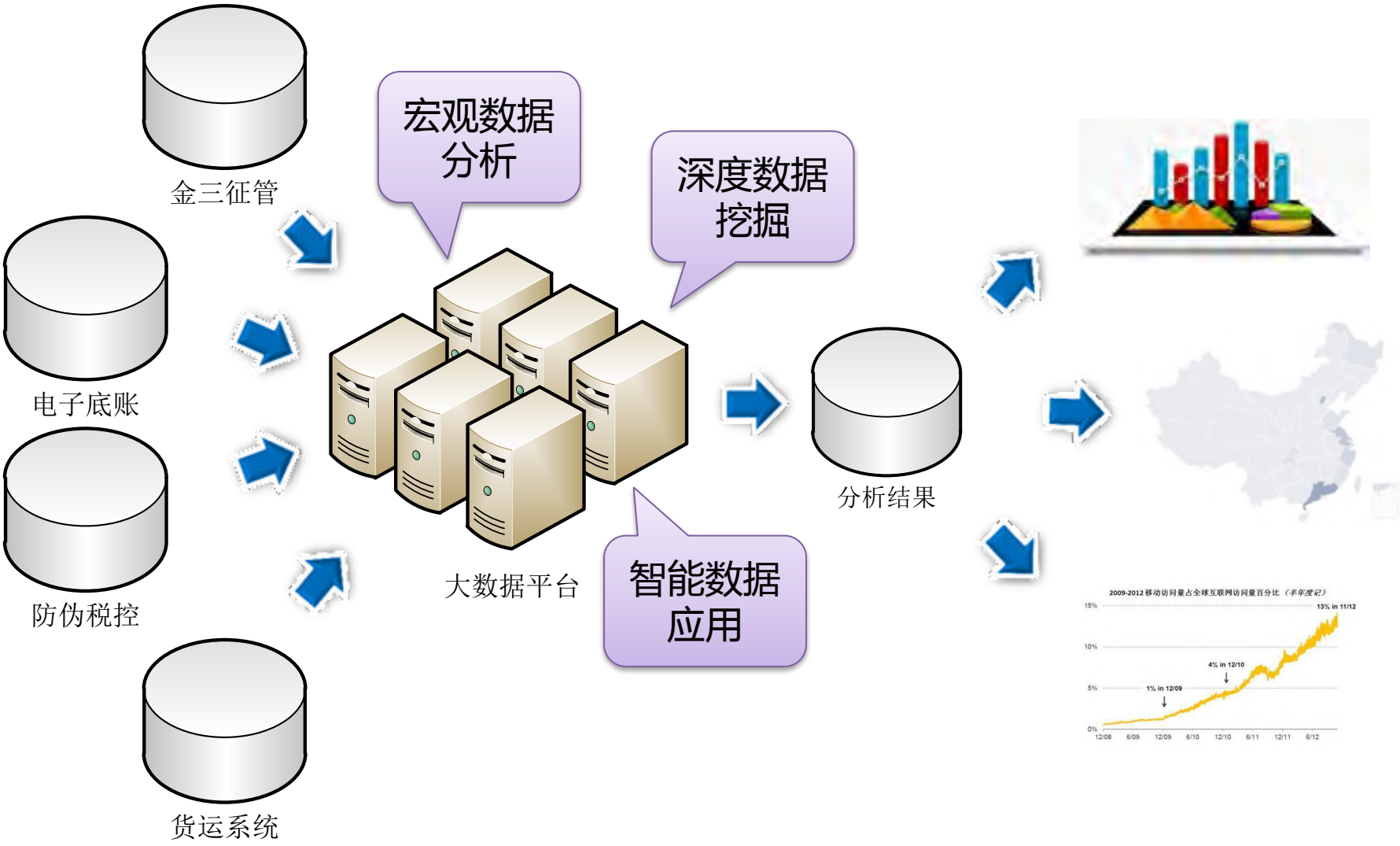
支撑海量的数据及流量：对于大型互联网应用来说，需要应对PB级的数据量和百万级的用户访问流量

低延时读写速度：应用必须快速响应用户请求并有效提升系统吞吐量，才能极大地提升用户的满意度

大规模集群管理：为应对以上问题而构建大规模分布式集群

庞大的运营成本的考量：在构建集群时大幅度降低软硬件及人力投入

大数据转型的方向


金三征管
电子底账
防伪税控
货运系统
宏观数据分析
深度数据挖掘
智能数据应用
大数据平台
分析结果
2009-2012 移动访问量占全球互联网访问量百分比（半年度记）
13% in 11/12
4% in 12/10
1% in 12/09
15%
10%
5%
0%
12/08
6/09
12/09
6/10
12/10
6/11
12/11
6/12

如何将传统业务系统向大数据转型

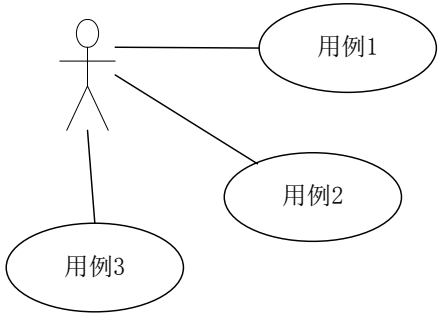
- 一边技术研究，一边设计开发，还要保证质量
- 在传统业务系统上改造，既要平稳，还要快速

我们面临的状况 1

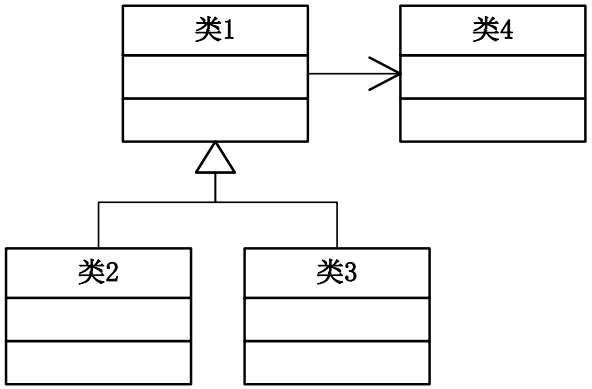
1. 快速地软件开发
2. 快速解决了技术难题
3. 快速实现了业务功能
4. 却产生了一堆难于维护的代码

我们面临的状况 2

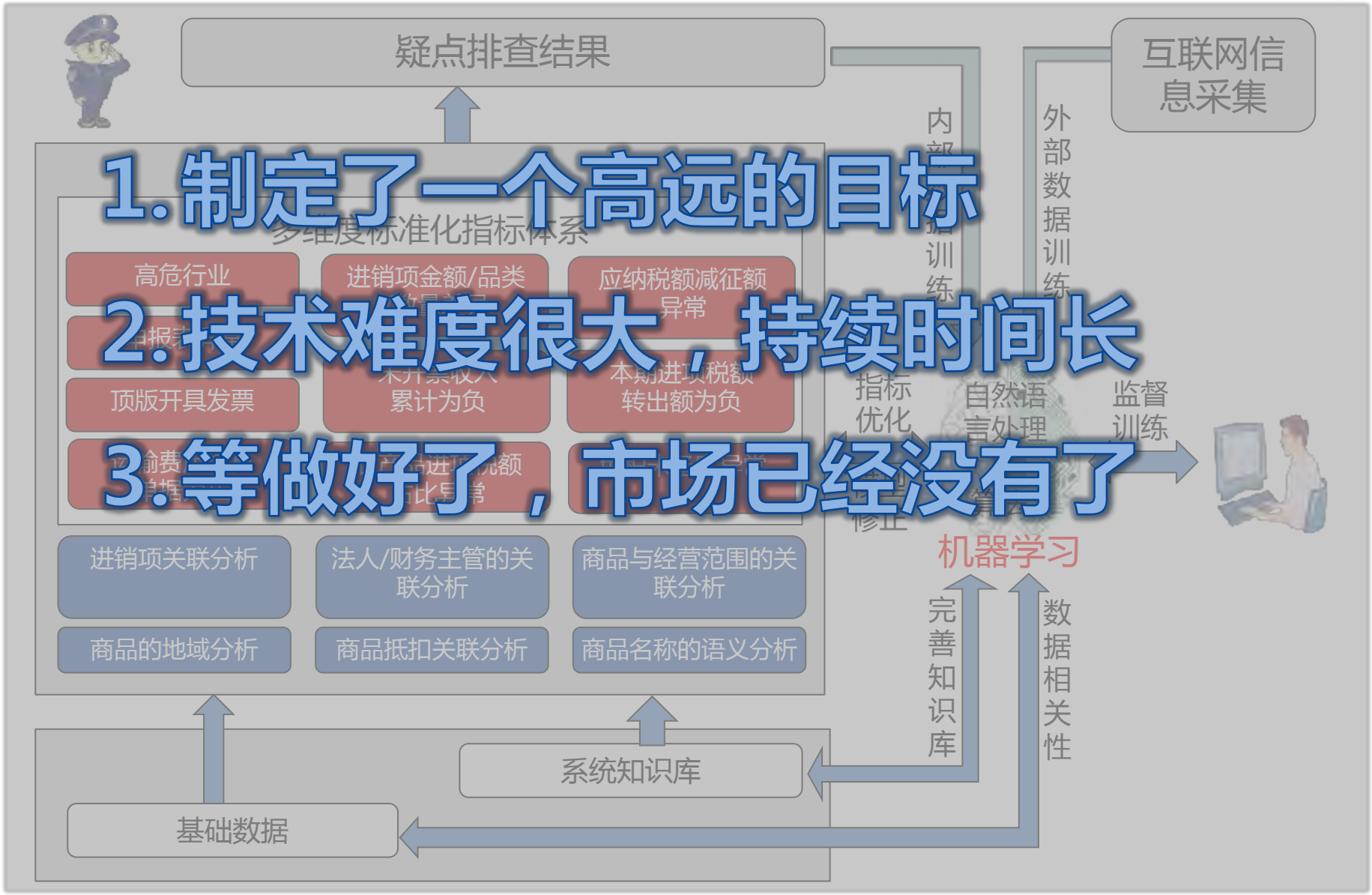
1. 在原有的基础上改造
2. 代码臃肿，可读性差
3. 技术不成熟，软件BUG多
4. 设计开发过程中的运行调试成本巨大




```
import org.apache.spark.SparkConf

object Csv {
  def main(args: Array[String]) {
    val conf = (new SparkConf).setMaster("local").setAppName("Csv");
    val sc = new SparkContext(conf)
    val input = "file:///d:\\input\\aa.csv"//args(0)
    val lines = sc.wholeTextFiles(input)
    val result = lines.flatMap { case(_, x) =>
      val reader = new CSVReader(new StringReader(x))
      val rowarray = Array[Array[String]]()
      def readnext(reader: CSVReader) {
        val row = reader.readNext()
        rowarray+row
        if(row.isEmpty) {}
        else readnext(reader)
      }
      rowarray
    }
    result.foreach { row => row.foreach { x => print(x) } }
  }
}
```



我们面临的状况 3



1. 制定了一个高远的目标
2. 技术难度很大，持续时间长
3. 等做好了，市场已经没有了

TiD2016
质量竞争力大会
软件研发顶级盛会

下一代软件研发
SOFTWARE DEVELOPMENT
更高 更新 更深

我们的实践

一次大数据改造过程
从传统BI项目开始
逐步开展大数据业务

项目背景

- 我们做了基于税务数据的分析多年
  - 纳税评估
  - 风险控制
  - 绩效考核

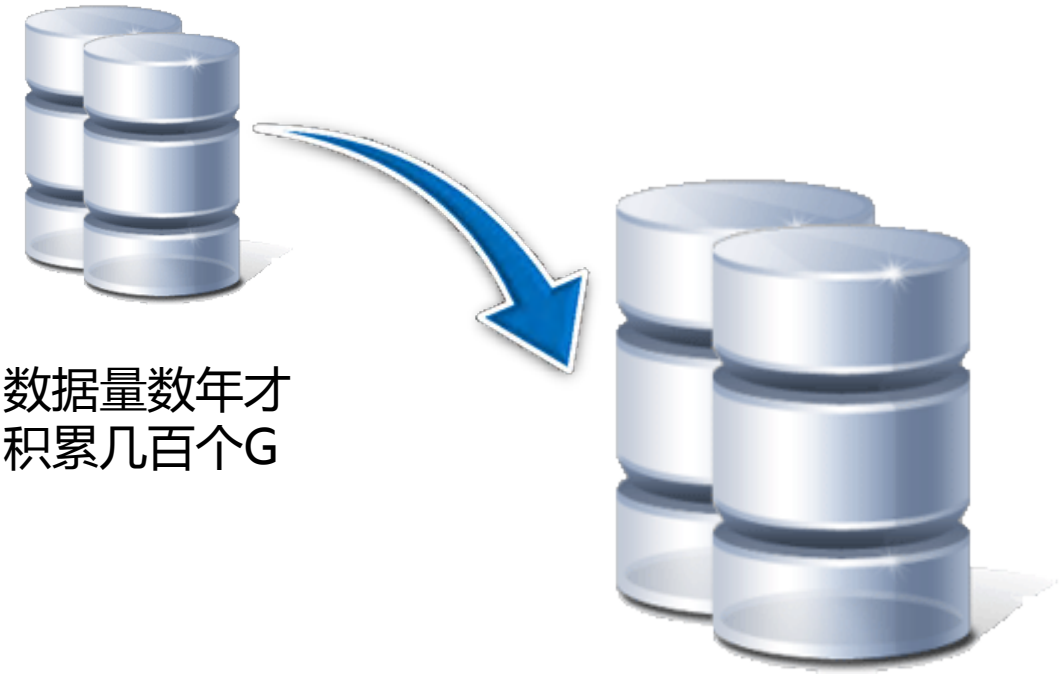
- 我们有一个基于增值税分析的BI系统
  - 对增值税发票来源地与流向地分析
  - 对增值税虚开票进行风险控制
  - 从行业与地域等维度进行分析统计

增值税发票来源地/流向地图表展示

互联网转型为该项目带来新的机遇

- 网络实时开票
  - 可以更加实时地获得开票数据
  - 实现实时的数据分析与展现
- 电子底账
  - 更加准确与实时地获得跨省数据
  - 实现跨省的发票虚开风控管理
- 营改增项目
  - 可以获得更加全面的数据
  - 更加全面地展现各地区的经济状况

互联网转型为该项目带来新的挑战



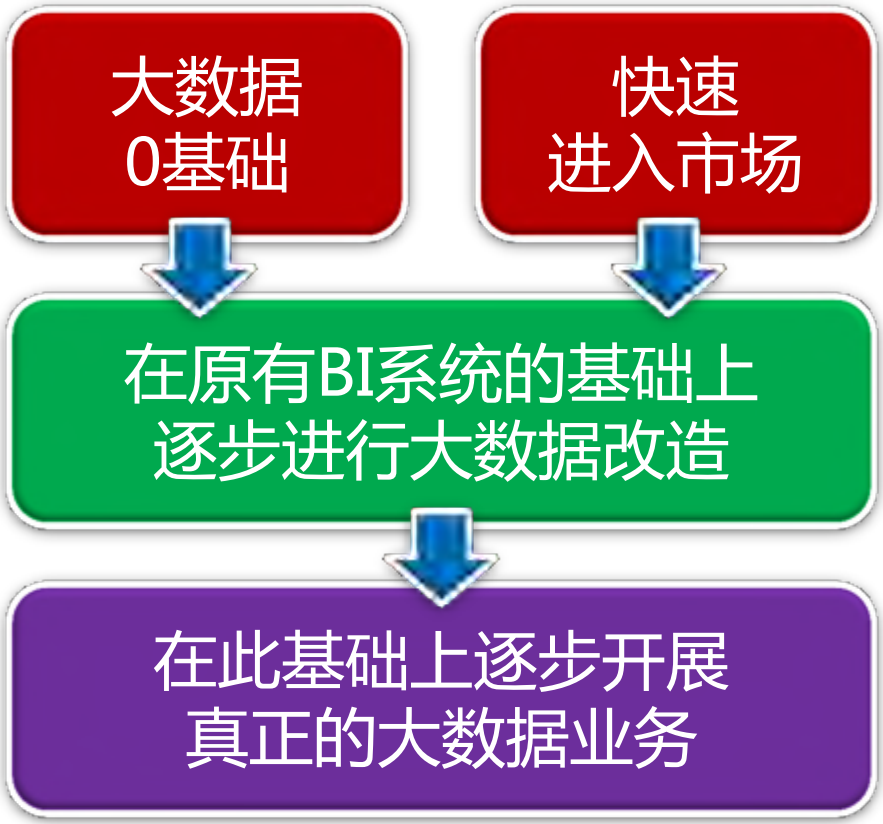
数据量数年才
积累几百个G

数据增量
1年就几个T

- 数据存储成本越来越大
- 数据查询速度越来越慢
- 数据扩展难度陡然增加

大数据转型

我们面临的难题与应对策略

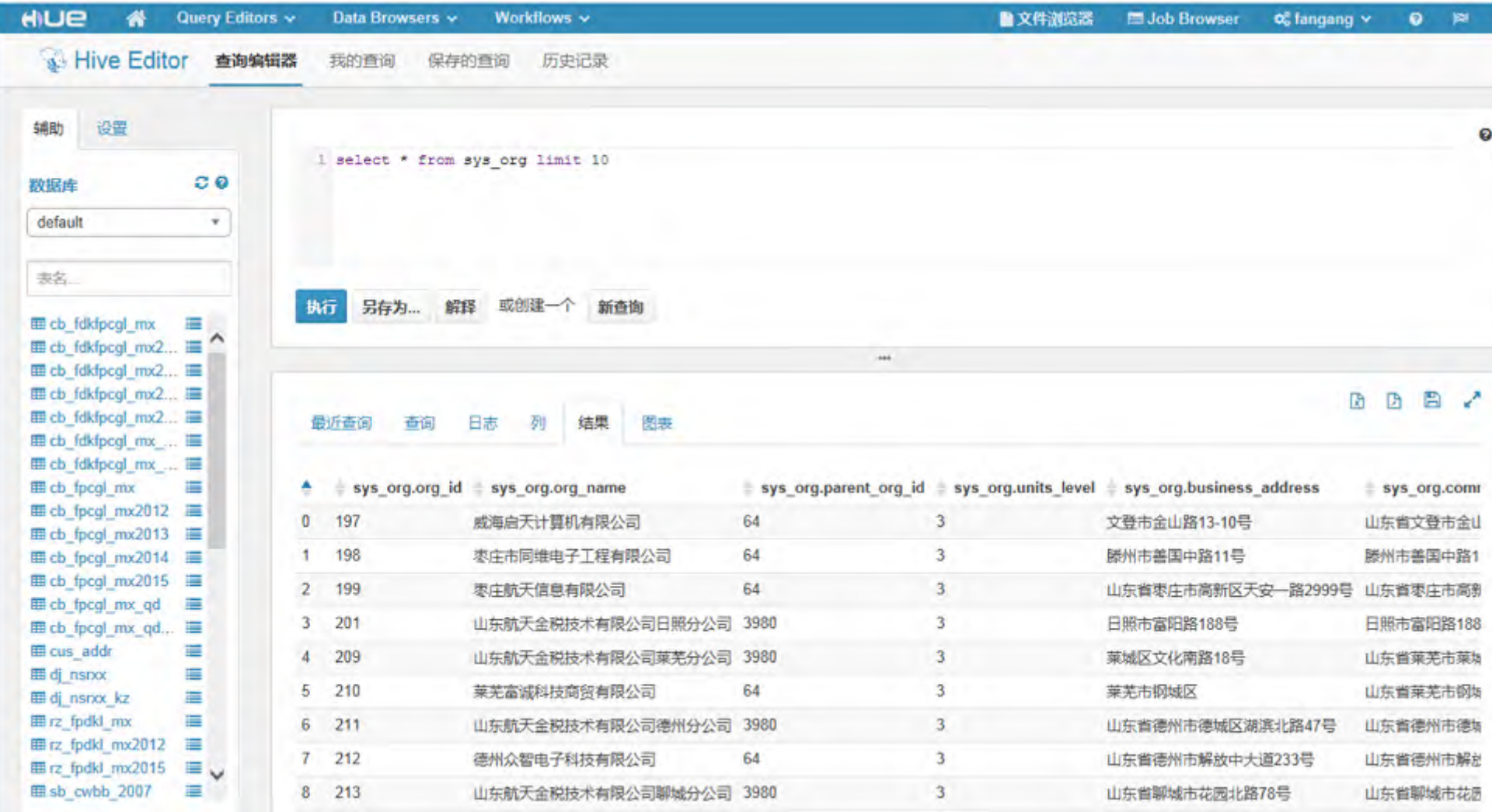


选择正确的技术：Hive+SparkSQL

- 大数据开发的难题
  - 开发人员不熟悉Spark/scala编程模式
  - 开发人员不熟悉分布式计算的设计原理
  - 设计与开发分布式计算平台很麻烦
  - 需要大量的业务数据需要分析与处理
  - 开发人员数据SQL语句
- 解决方案
  - 提供一个开发平台可以使用SQL语句编写分布式数据分析与处理程序
  - 可以将写好的SQL语句转换成Spark程序

Hive在Hue中进行数据查询

TiD2016



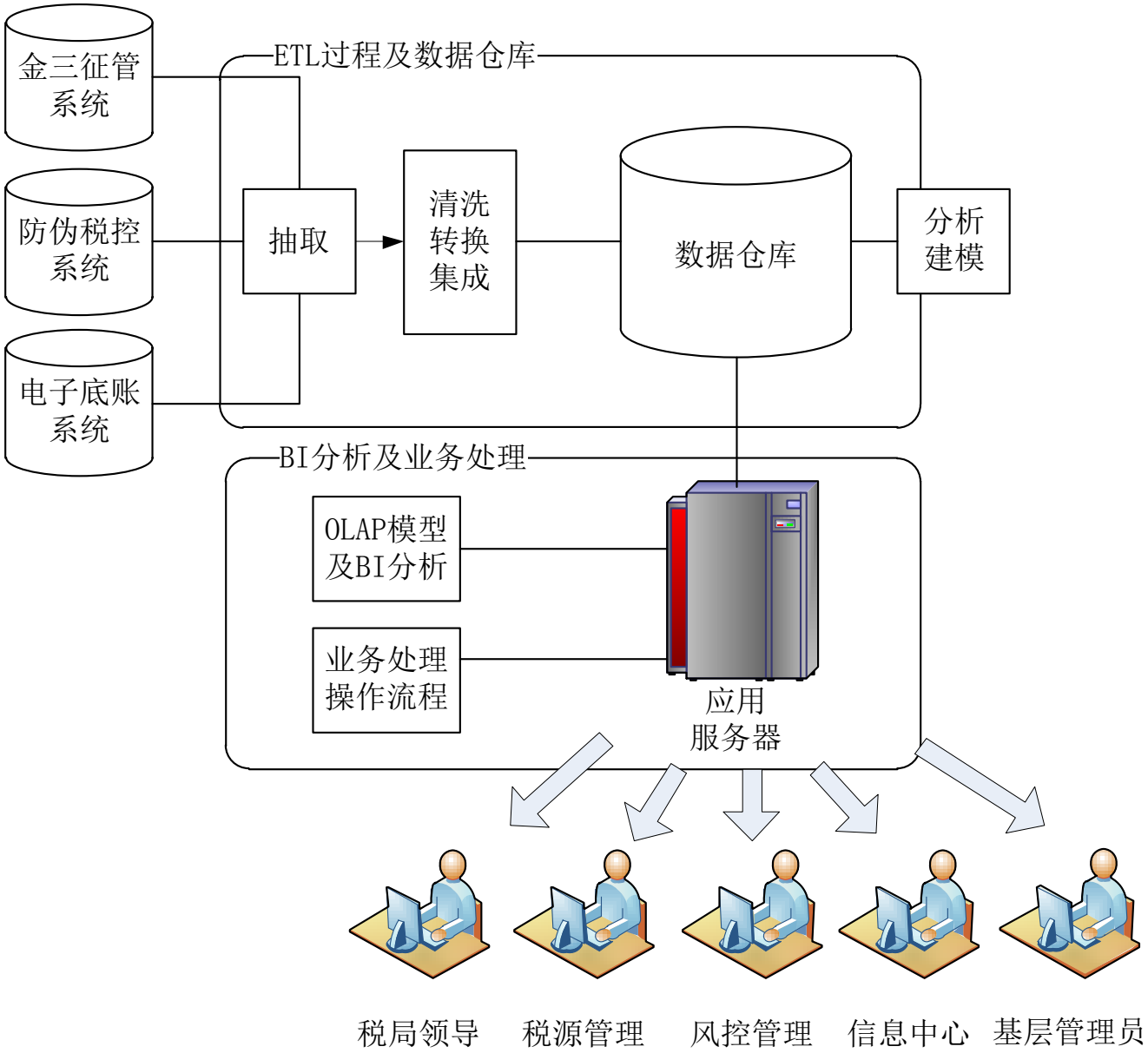
	sys_org.org_id	sys_org.org_name	sys_org.parent_org_id	sys_org.units_level	sys_org.business_address	sys_org.comm
0	197	威海启天计算机有限公司	64	3	文登市金山路13-10号	山东省文登市金山
1	198	枣庄市同维电子工程有限公司	64	3	滕州市善国中路11号	滕州市善国中路1
2	199	枣庄航天信息有限公司	64	3	山东省枣庄市高新区天安一路2999号	山东省枣庄市高新
3	201	山东航天金税技术有限公司日照分公司	3980	3	日照市富阳路188号	日照市富阳路188
4	209	山东航天金税技术有限公司莱芜分公司	3980	3	莱城区文化南路18号	山东省莱芜市莱城
5	210	莱芜富诚科技商贸有限公司	64	3	莱芜市钢城区	山东省莱芜市钢城
6	211	山东航天金税技术有限公司德州分公司	3980	3	山东省德州市德城区湖滨北路47号	山东省德州市德城
7	212	德州众智电子科技有限公司	64	3	山东省德州市解放中大道233号	山东省德州市解放
8	213	山东航天金税技术有限公司聊城分公司	3980	3	山东省聊城市花园北路78号	山东省聊城市花园

将验证后的Hive语句贴入SparkSQL

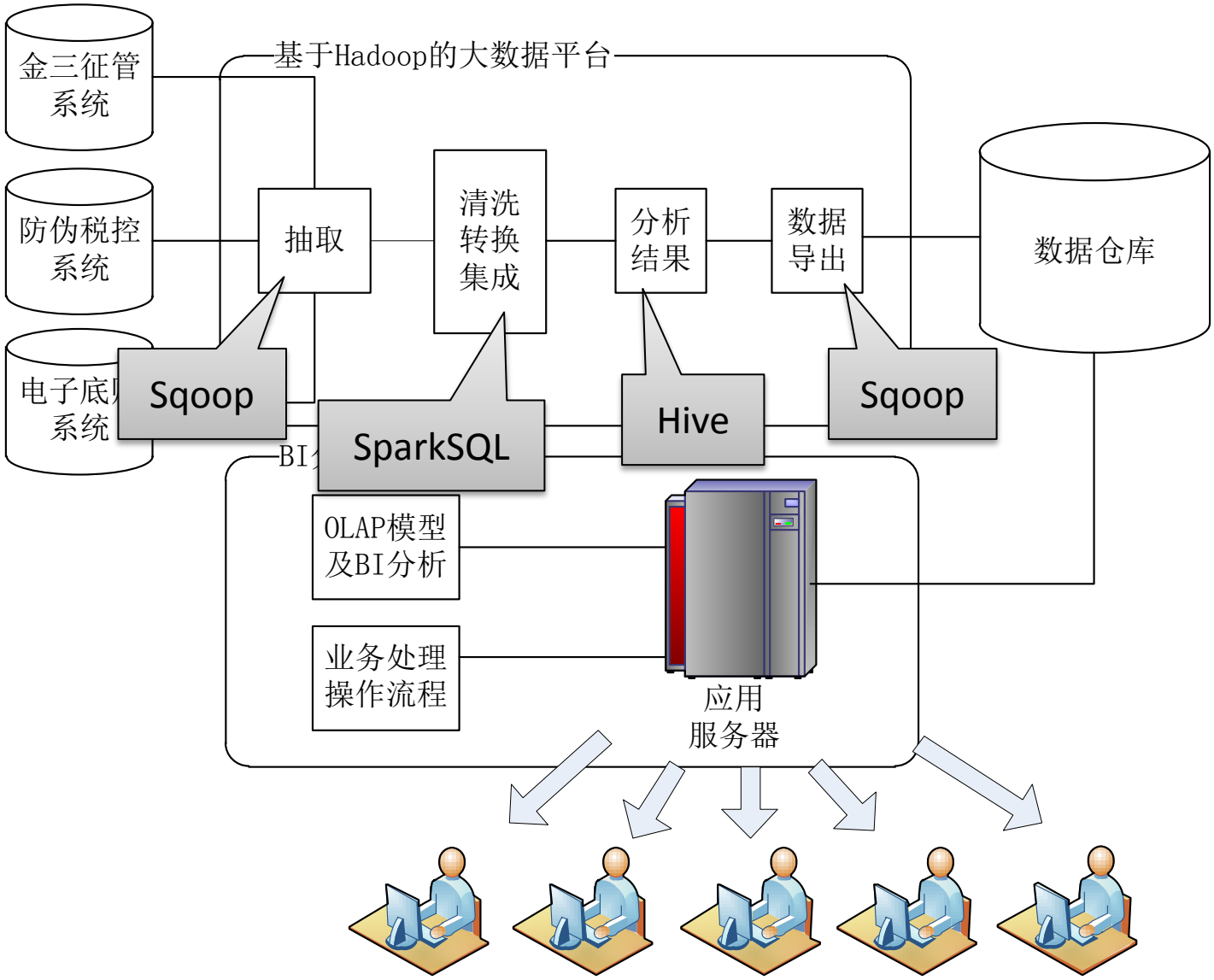
TiD2016

```
/**
 * 进项分析按纳税人聚合
 * @author hada
 */
object JxfxNsr {
  def main(args: Array[String]) {
  val task = LogUtil.start("dw.agg.jxfxNsr")
  try {
    val sc = SparkUtil.init("dw.agg.jxfxNsr")
    val hc = SparkUtil.getSqlContext(sc)
    hc.udf.register("getDateKey", (dateString:String) =>
         DateUtil.format(DateUtil.getTime(dateString),"yyyyMM").toLong)

    val result = hc.sql("SELECT getDateKey(MX.KPRQ) DATE_KEY,MX.GF_NSRSBH GF_NSR_KEY , "+
      " MX.GF_SWJG_DM GF_SWJG_KEY,NSR.NSRSBH XF_NSR_KEY,MX.XF_SWJG_DM XF_SWJG_KEY,MX.FP_LB,"+
      " SUM(QD.WP_SL) WP_SL,SUM(QD.JE) JE,SUM(QD.SE) SE,COUNT(DISTINCT MX.JXFP_ID) FPFS "+
      " FROM etl.ETL_JXFP MX INNER JOIN etl.ETL_JXFP_QD QD ON MX.JXFP_ID = QD.JXFP_ID "+
      " INNER JOIN dw.DW_DM_NSR NSR ON MX.XF_NSRSBH = NSR.NSR_KEY"+
      " WHERE (MX.ZFBZ='N' or MX.ZFBZ is null)"+
      " GROUP BY getDateKey(MX.KPRQ) , MX.GF_NSRSBH, MX.GF_SWJG_DM,NSR.NSRSBH,"+
      " MX.XF_SWJG_DM,MX.FP_LB,MX.XF_NSRMC,QD.SL")

    DataFrameUtil.save(result, "dw", "dw_agg_jxfx_nsr")
    LogUtil.end(task)
    } catch { case ex:Exception  => LogUtil.error(task, ex) }
  }
}
```

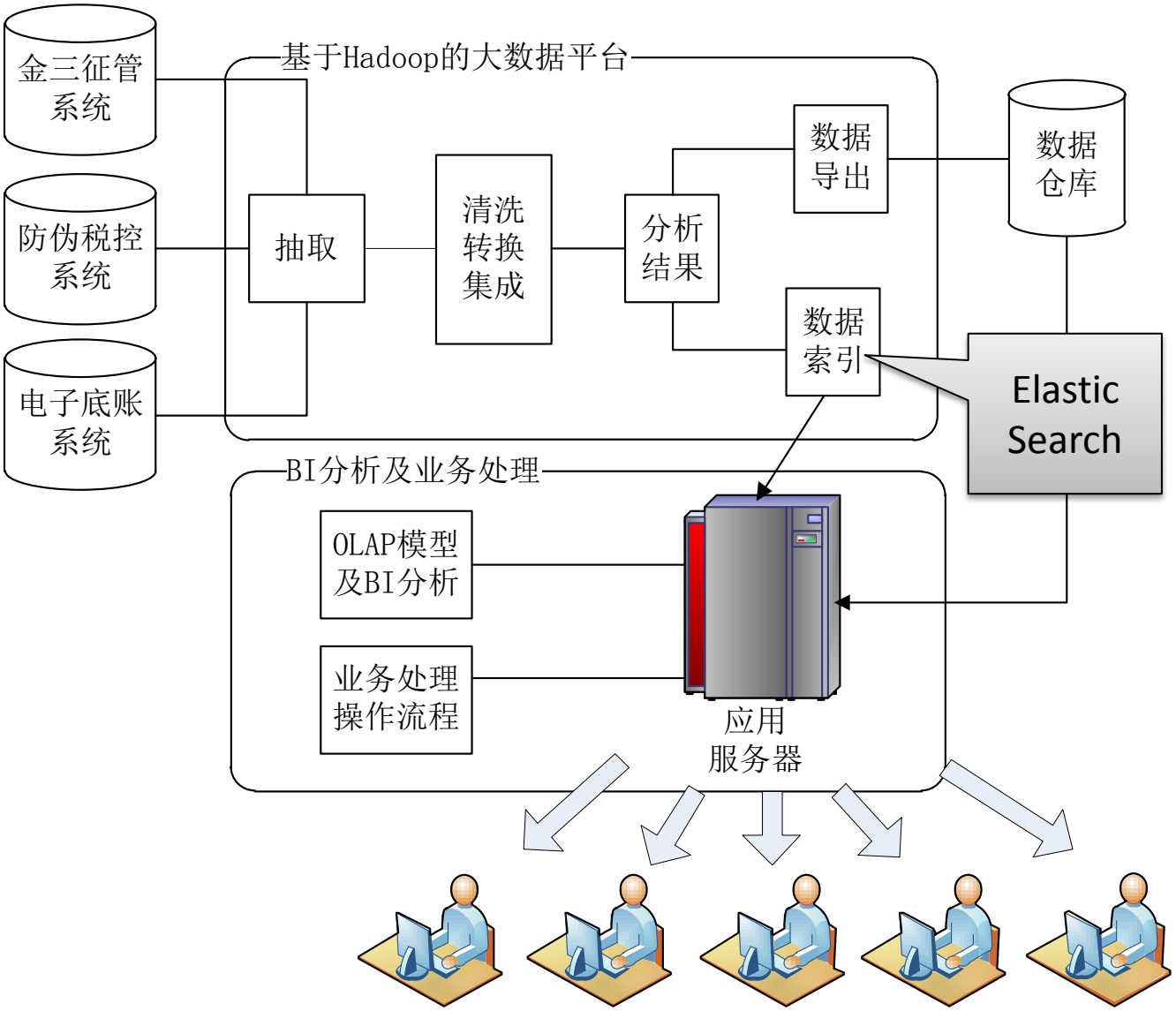
制定合理的项目目标与技术方案



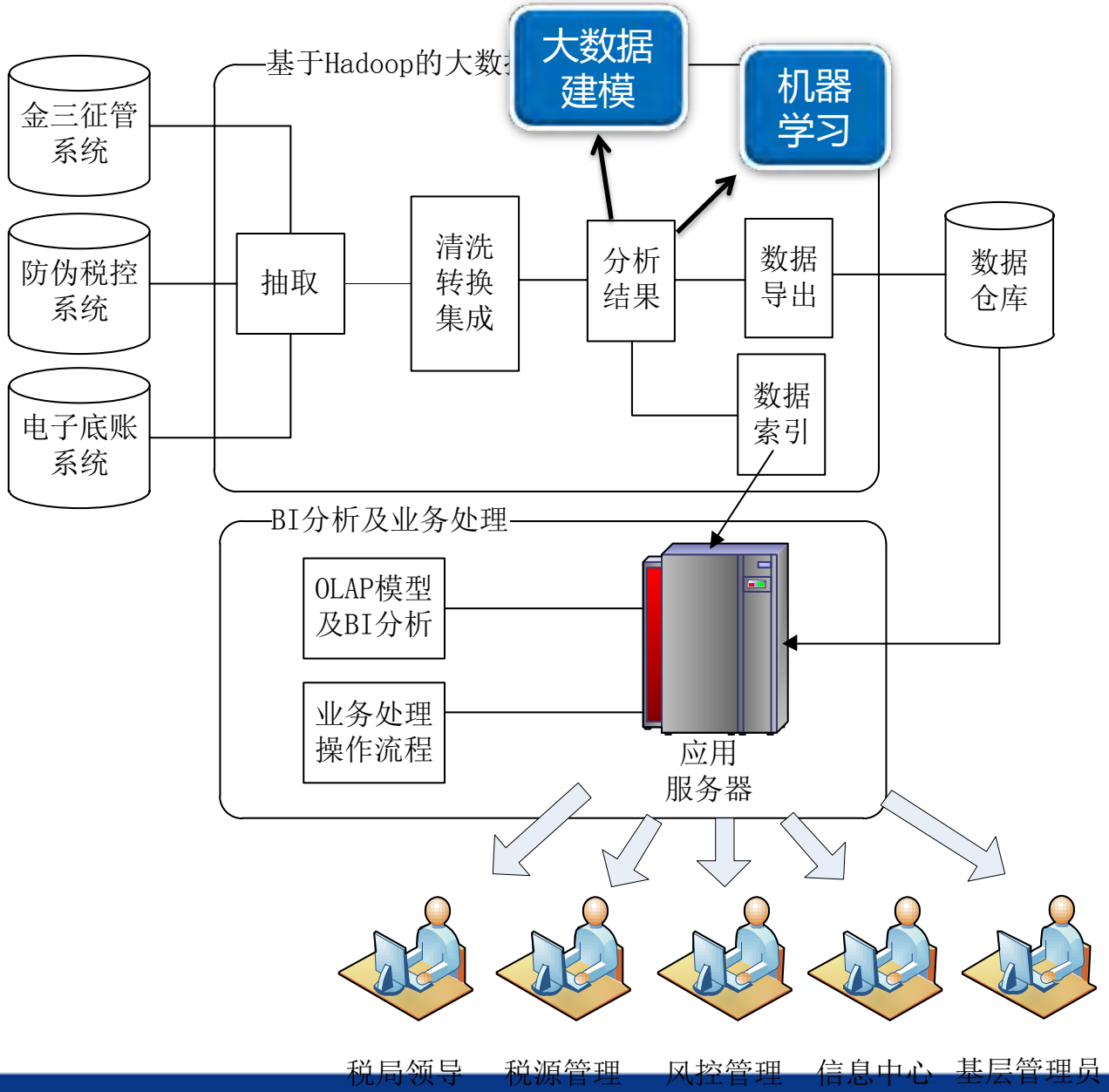
第一阶段改造


金三征管系统
防伪税控系统
电子底账系统
基于Hadoop的大数据平台
抽取
清洗转换集成
分析结果
数据导出
数据仓库
Sqoop
SparkSQL
Hive
Sqoop
BI
OLAP模型及BI分析
业务处理操作流程
应用服务器
税局领导
税源管理
风控管理
信息中心
基层管理员

第二阶段改造


金三征管系统
防伪税控系统
电子底账系统
基于Hadoop的大数据平台
抽取
清洗转换集成
分析结果
数据导出
数据索引
数据仓库
Elastic Search
BI分析及业务处理
OLAP模型及BI分析
业务处理操作流程
应用服务器
税局领导
税源管理
风控管理
信息中心
基层管理员

第三阶段改造


大数据建模
机器学习
基于Hadoop的大数
金三征管系统
防伪税控系统
电子底账系统
抽取
清洗转换集成
分析结果
数据导出
数据仓库
数据索引
BI分析及业务处理
OLAP模型及BI分析
业务处理操作流程
应用服务器
税局领导
税源管理
风控管理
信息中心
基层管理员

TiD2016
质量竞争力大会
软件研发顶级盛会

下一代软件研发
SOFTWARE DEVELOPMENT
更高 更新 更深

小步快跑的开发模式

技术解决了代码很烂
代码写得好调试很麻烦

关于设计的思考...

- 技术解决了但代码写得很烂
  - 对新技术不甚熟悉，需反复修改调试
  - 为了便于修改调试，只好顺序编程
  - 技术问题解决了，代码却难于维护

- 代码写得好却难于调试改错
  - 代码分层分模块十分优雅
  - 代码出错却难于定位问题
  - 为解决问题需大范围修改代码

问题的关键在于：

顺序编程的烦恼


```
/**
 * 增值税进项发票ETL程序，从增值税专用发票表(DZDZ_FPXX_ZZSFP)中抽取数据，进行集成、清洗、转换操作，
 * @author Fangang
 */
object ZzsfpJx {
  def main(args: Array[Stri
    val conf = (new SparkConf).setMaster("master").setAppName("zzsfpJx");
    val sc = new SparkContex
    val hc = new HiveContext

    val dateFormat =  new Simp                    yyyy-MM-dd HH:mm:ss")
    val now = dateFormat.format(new Date())

    hc.udf.register("getJxfpId                    String, kprq:Date) =>
         if(null==kprq) fpdm+"                    "X"+fphm+"X"+kprq)
    hc.udf.register("fillZero", (swjg:String)=> if(null==swjg) swjg else swjg+"00")

    val result = hc.sql("SELECT                   ,D.KPRQ) JXFP_ID,D.FPDM,D.FPHM,"+
      "'YB' FP_LB,D.JE JE,D.SE/D                  XF_NSRSBH,D.XFMC XF_NSRMC,"+
      s"D.GFSBH GF_NSRSBH,D.GFMC                  L_BDRQ RZSJ,D.BSSWJG_DM SWJG_DM,'$now
      "fillZero(D.GF_QXSWJG_DM) GF_SWJG_DM,fillZero(D.XF_QXSWJG_DM) XF_SWJG_DM,D.ZFBZ,D.SKM
      "FROM dzdz.DZDZ_FPXX_ZZSFP D

    val path = "/user/hive/warehou
    result.save(path, SaveMode.Overwrite)
  }
}
```

没有分层

没有分模块

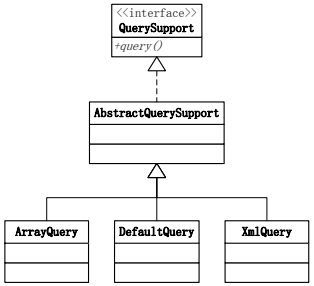
没有代码复用

代码臃肿

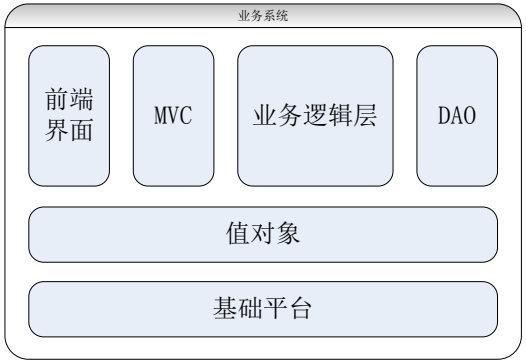
无法功能扩展

分层、分模块...

需求分析



全面的设计



分层分模块

开发编码

集成组装

运行调试

小步快跑的开发模式

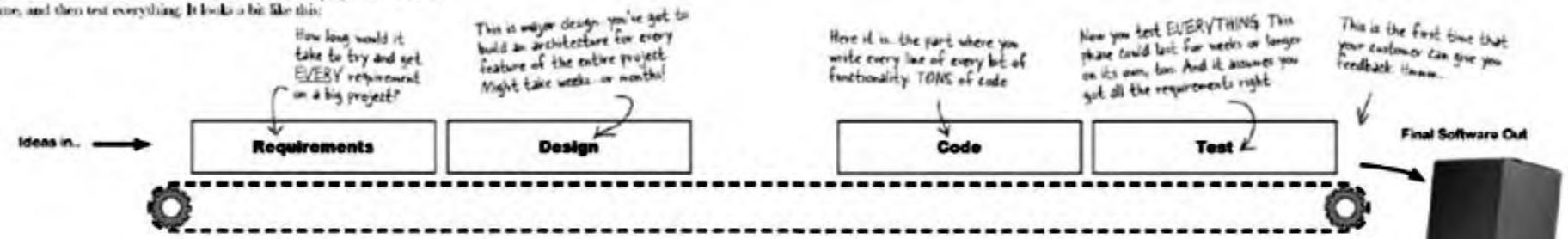
- 用最快的速度开发一个最核心的功能
- 让第一个版本运行起来并可以验证
- 在第一个版本的基础上不断添加新功能
- 每次只添加一个很简单、很单一的功能
- 对添加的功能进行验证并使其可运行
- 每个开发周期只有10分钟到半小时
- 每次添加新功能时只进行恰如其分地设计
- 下次添加新功能时如果需要应当适时进行重构
- 复杂的系统是由一次次正确的开发积累而成

小步快跑的开发模式

Each iteration is a mini-project

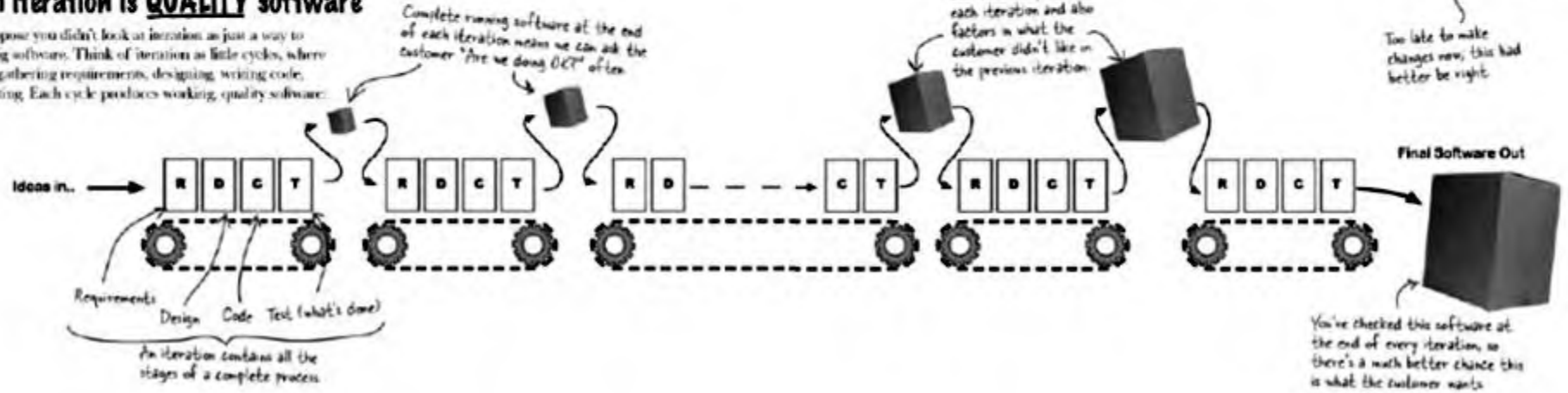
With iteration, you take the steps you'd follow to build the entire project, and put those steps into **each iteration**. In fact, each iteration is a mini-project, with its own requirements, design, coding, testing, etc., built right in. So you're not showing your customer junk... you're showing them well-developed bits of the final software.

Think about how most software is developed: You gather requirements (what your customer wants), build a design for the entire project, code for a long time, and then test everything. It looks a bit like this:



Each iteration is QUALITY software

But suppose you didn't look at iteration as just a way to write big software. Think of iteration as little cycles, where you're gathering requirements, designing, writing code, and testing. Each cycle produces working, quality software:



TiD2016
质量竞争力大会
软件研发顶级盛会

下一代软件研发
SOFTWARE DEVELOPMENT
更高 更新 更深

SparkSQL的代码优化过程

第一个版本的实现

TiD2016

```
/**
 * 增值税进项发票ETL程序，从增值税专用发票表(DZDZ_FPXX_ZZSFP)中抽取数据，进行集成、清洗、转换操作，
 * @author Fangang
 */
object ZzsfpJx {
  def main(args: Array[String]) {
    val conf = (new SparkConf).setMaster("master").setAppName("zzsfpJx");
    val sc = new SparkContext(conf)
    val hc = new HiveContext(sc)

    val dateFormat =  new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")
    val now = dateFormat.format(new Date())

    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
         if(null==kprq) fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)
    hc.udf.register("fillZero", (swjg:String)=> if(null==swjg) swjg else swjg+"00")

    val result = hc.sql("SELECT getJxfpId(D.FPDM,D.FPHM,D.KPRQ) JXFP_ID,D.FPDM,D.FPHM,"+
      "'YB' FP_LB,D.JE JE,D.SE/D.JE SL,D.SE SE,D.XFSBH XF_NSRSBH,D.XFMC XF_NSRMC,"+
      s"D.GFSBH GF_NSRSBH,D.GFMC GF_NSRMC,D.KPRQ,D.RZDKL_BDRQ RZSJ,D.BSSWJG_DM SWJG_DM,'$now
      "fillZero(D.GF_QXSWJG_DM) GF_SWJG_DM,fillZero(D.XF_QXSWJG_DM) XF_SWJG_DM,D.ZFBZ,D.SKM
      "FROM dzdz.DZDZ_FPXX_ZZSFP D ")

    val path = "/user/hive/warehouse/etl.db/etl_jxfp"
    result.save(path, SaveMode.Overwrite)
  }
}
```

开发下一个功能

TiD2016

```
/**
 * 增值税进项清单发票ETL程序，从增值税专用发票表(DZDZ_HWXX_ZZSFP)和进项发票表(ETL_JXFP)中抽取数据，
 * @author Liruiming
 */
object ZzsfpJxQd {
  def main(args: Array[String]) {
    val conf = (new SparkConf).setMaster("master").setAppName("zzsfpJx");
    val sc = new SparkContext(conf)
    val hc = new HiveContext(sc)
    val dateFormat =  new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")
    val now = dateFormat.format(new Date())
    hc.udf.register("getJxfpqdId", (fpdm:String, fphm:String, kprq:Date,
      qdbz:String, id:Double) => if(null==kprq) fpdm+"X"+fphm+"X"+"null"+"X"+qdbz+"X"+id
         else fpdm+"X"+fphm+"X"+kprq+"X"+qdbz+"X"+id)
    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
      if (null==kprq )fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)

    val result = hc.sql("SELECT getJxfpqdId(D.FPDM,D.FPHM,R.KPRQ,D.QDBZ,D.ID) JXFPQD_ID,"+
        "getJxfpId(D.FPDM,D.FPHM,R.KPRQ) JXFP_ID ,D.ID HH,'YB' FP_LB,D.HWMC WP_MC,"+
        "D.DW WP_DW,D.GGXH WP_XH,D.SL WP_SL,D.DJ DJ,D.JE JE,D.SLV SL,"+
       s"D.SE SE,R.RZSJ RZSJ,'$now' CZSJ,R.KPRQ KPRQ,D.QDBZ "+
        "FROM dzdz.DZDZ_HWXX_ZZSFP D JOIN etl.ETL_JXFP R "+
        "ON (D.FPDM = R.FPDM AND D.FPHM = R.FPHM)")
    val path = "/user/hive/warehouse/etl.db/etl_jxfp_qd"
    result.save(path, SaveMode.Overwrite)
  }
}
```

第一个版本的分析

TiD2016

```
/**
 * 增值税进项发票ETL程序，从增值税专用发票表(DZDZ_FPXX_ZZSFP)中抽取数据，进行集成、清洗、转换操作，
 * @author Fangang
 */
object ZzsfpJx {
  def main(args: Array[String]) {
    val conf = (new SparkConf).setMaster("master").setAppName("zzsfpJx");
    val sc = new SparkContext(conf)
    val hc = new HiveContext(sc)
    val dateFormat =  new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")
    val now = dateFormat.format(new Date())
    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
         if(null==kprq) fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)
    hc.udf.register("fillZero", (swjg:String)=> if(null==swjg) swjg else swjg+"00")

    val result = hc.sql("SELECT getJxfpId(D.FPDM,D.FPHM,D.KPRQ) JXFP_ID,D.FPDM,D.FPHM,"+
      "'YB' FP_LB,D.JE JE,D.SE/D.JE SL,D.SE SE,D.XFSBH XF_NSRSBH,D.XFMC XF_NSRMC,"+
      s"D.GFSBH GF_NSRSBH,D.GFMC GF_NSRMC,D.KPRQ,D.RZDKL_BDRQ RZSJ,D.BSSWJG_DM SWJG_DM,'$now'"+
      "fillZero(D.GF_QXSWJG_DM) GF_SWJG_DM,fillZero(D.XF_QXSWJG_DM) XF_SWJG_DM,D.ZFBZ,D.SKM"+
      "FROM dzdz.DZDZ_FPXX_ZZSFP D ")
    val path = "/user/hive/warehouse/etl.db/etl_jxfp"
    result.save(path, SaveMode.Overwrite)
  }
}
```

SparkUtil

DateUtil

DataFrameUtil

第二个版本进行重构

TiD2016

```
/**
 * 增值税进项发票ETL程序，从增值税专用发票表(DZDZ_FPXX_ZZSFP)中抽取数据，进行集成、清洗
 * @author Fangang
 */
object ZzsfpJx {
  def main(args: Array[String]) {
    val sc = SparkUtil.init("zzsfpJx")
    val hc = SparkUtil.getSqlContext(sc)
    val now = DateUtil.getNow

    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
        if(null==kprq) fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)
    hc.udf.register("fillZero", (swjg:String)=> if(null==swjg) swjg else swjg+"00

    val result = hc.sql("SELECT getJxfpId(D.FPDM,D.FPHM,D.KPRQ) JXFP_ID,D.FPDM,D.
      "'YB' FP_LB,D.JE JE,D.SE/D.JE SL,D.SE SE,D.XFSBH XF_NSRSBH,D.XFMC XF_NSRMC,
      s"D.GFSBH GF_NSRSBH,D.GFMC GF_NSRMC,D.KPRQ,D.RZDKL_BDRQ RZSJ,D.BSSWJG_DM SW
      "fillZero(D.GF_QXSWJG_DM) GF_SWJG_DM,fillZero(D.XF_QXSWJG_DM) XF_SWJG_DM,D.
      "FROM dzdz.DZDZ_FPXX_ZZSFP D ")

    DataFrameUtil.save(result, "etl", "etl_jxfp")
  }
}
```

第二个版本进行重构

TiD2016

```
/**
 * 增值税进项清单发票ETL程序，从增值税专用发票表(DZDZ_HWXX_ZZSFP)和进项发票表(ETL_JXFP)中抽取数据，
 * @author Liruiming
 */
object ZzsfpJxQd {
  def main(args: Array[String]) {
    val sc = SparkUtil.init("zzsfpJxQd")
    val hc = SparkUtil.getSqlContext(sc)
    val now = DateUtil.getNow

    hc.udf.register("getJxfpqdId", (fpdm:String, fphm:String, kprq:Date,
      qdbz:String, id:Double) => if(null==kprq) fpdm+"X"+fphm+"X"+"null"+"X"+qdbz+"X"+id
          else fpdm+"X"+fphm+"X"+kprq+"X"+qdbz+"X"+id)
    hc.udf.register("getJxfpId", (fpdm:String, fphm:String, kprq:Date) =>
      if (null==kprq )fpdm+"X"+fphm+"X" else fpdm+"X"+fphm+"X"+kprq)

    val result = hc.sql("SELECT getJxfpqdId(D.FPDM,D.FPHM,R.KPRQ,D.QDBZ,D.ID) JXFPQD_ID,"+
        "getJxfpId(D.FPDM,D.FPHM,R.KPRQ) JXFP_ID ,D.ID HH,'YB' FP_LB,D.HWMC WP_MC,"+
        "D.DW WP_DW,D.GGXH WP_XH,D.SL WP_SL,D.DJ DJ,D.JE JE,D.SLV SL,"+
       s"D.SE SE,R.RZSJ RZSJ,'$now' CZSJ,R.KPRQ KPRQ,D.QDBZ "+
        "FROM dzdz.DZDZ_HWXX_ZZSFP D JOIN etl.ETL_JXFP R "+
        "ON (D.FPDM = R.FPDM AND D.FPHM = R.FPHM)")

    DataFrameUtil.save(result, "etl", "etl_jxfp_qd")
  }
}
```

底层的设计变更


```
object DataFrameUtil {
  val targetDir = PropertyFile.getProperty("targetDir")
  /**
   * create hive table
   * @param schema
   * @param tableName
   */
  def createTable(sqlContext: SQLContext, schema: String, tableName: String) {
    val hql = HQLFactory.getHQL(schema, tableName)
    sqlContext.sql(hql)
  }
  /**
   * save data after the ETL process,
   * @param result
   * @param path
   */
  def save(data: DataFrame, path: String) {
      data.save(path, SaveMode.Overwrite)
  }
  /**
   * save data after the ETL process,
   * @param data
   * @param schema
   * @param table
   */
  def save(data: DataFrame, schema: String, table: String) {
    val path = targetDir+"/"+schema+".db/"+table
    if(!FileUtil.isExists(path)) createTable(data.sqlContext, schema, table)
    save(data, path)
  }
}
```

底层的设计变更


```
object DataFrameUtil {
  val targetDir = PropertyFile.getProperty("targetDir")
  /**
   * create hive table
   * @param schema
   * @param tableName
   */
  def createTable(sqlContext: SQLContext, schema: String, tableName: String) {
    val hql = HQLFactory.getHQL(schema, tableName)
    sqlContext.sql(hql)
  }
  /**
   * save data after the ETL process,
   * @param result
   * @param path
   */
  def save(data: DataFrame, path: String) {
      data.save(path, SaveMode.Overwrite)
  }
  /**
   * save data after the ETL process,
   * @param data
   * @param schema
   * @param table
   */
  def save(data: DataFrame, schema: String, table: String) {
    val path = targetDir+"/"+schema+".db/"+table
    if(mode.equals(SaveMode.Overwrite)) dropTable(data.sqlContext, schema, table)
      createTable(data.sqlContext, schema, table)
    save(data, path, mode)  }
}
```

TiD2016
质量竞争力大会
软件研发顶级盛会

下一代软件研发
SOFTWARE DEVELOPMENT
更高 更新 更深

Sqoop的代码优化过程

代码复用
易于阅读
易于维护

第一个Sqoop程序


```
sqoop import --hive-import /
--connect jdbc:oracle:thin:@192.168.1.24:1521/SDHTXX /
--username admin --password-file /user/hive/.admin /
--query "select * from dzdz_fpxx_hyzp t where \$CONDITIONS" /
-m 10 --hive-database dzdz --hive-table dzdz_fpxx_hyzp /
--hive-overwrite /
--target-dir /user/hive/warehouse/dzdz.db/dzdz_fpxx_hyzp /
--outdir ../tmp --bindir ../tmp -z --direct --null-string '\\N'  /
--hive-delims-replacement " "
```

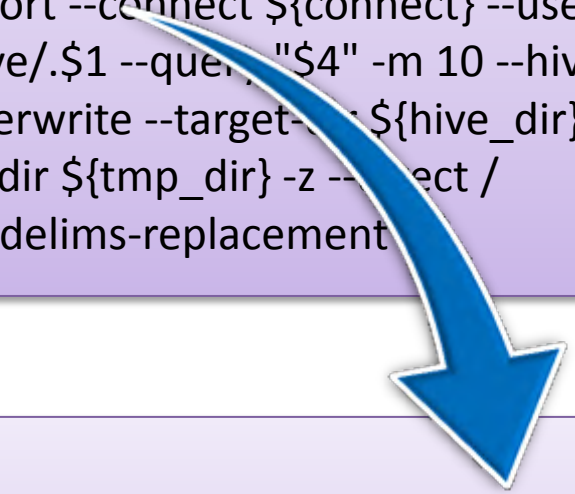
第二个Sqoop程序


```
sqoop import --hive-import /
--connect jdbc:oracle:thin:@192.168.1.24:1521/SDHTXX /
--username admin --password-file /user/hive/.admin /
--query "select * from dzdz_fpxx_zzsfp t where \$CONDITIONS" /
-m 10 --hive-database dzdz --hive-table dzdz_fpxx_zzsfp /
--hive-overwrite /
--target-dir /user/hive/warehouse/dzdz.db/dzdz_fpxx_zzsfp /
--outdir ../tmp --bindir ../tmp -z --direct --null-string '\\N'  /
--hive-delims-replacement " "
```

Sqoop的重构与整合

TiD2016

```
#!/bin/bash
cd /home/aisinobi/bin
connect=jdbc:oracle:thin:@192.168.1.24:1521/SDHTXX
hive_dir=/user/hive/warehouse
tmp_dir=../tmp
sqoop import --hive-import --connect ${connect} --username $1 /
--password-file /user/hive/.$1 --query "$4" -m 10 --hive-database $2 /
--hive-table $3 --hive-overwrite --target-dir ${hive_dir}/$2.db/$3 /
--outdir ${tmp_dir} --bindir ${tmp_dir} -z --direct /
--null-string '\\N'  --hive-delims-replacement
```

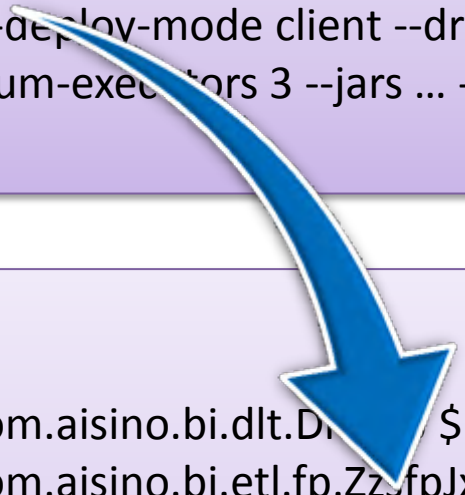


```
#!/bin/bash
sql="select * from dzdz_fpxx_hyzp t where kprq between
to_date('$1','yyyy-mm-dd') and to_date('$2 23:59:59','yyyy-mm-dd
hh24:mi:ss') and \$CONDITIONS"
map=SL=DOUBLE,SE=DOUBLE,JSHJ=DOUBLE
bash SqoopJdbc.sh DZDZ dzdz DZDZ_FPXX_HYZP "$sql" $map
```

Spark-submit的重构与整合

TiD2016

```
#!/bin/bash
if [ $# -lt 2 ]; then
  echo  "The usage is :  sparksubmit.sh classname jarfilename param1 ..."
else
  spark-submit --master yarn --deploy-mode client --driver-memory 10G /
    --executor-memory 10G --num-executors 3 --jars … --class $@
fi
```



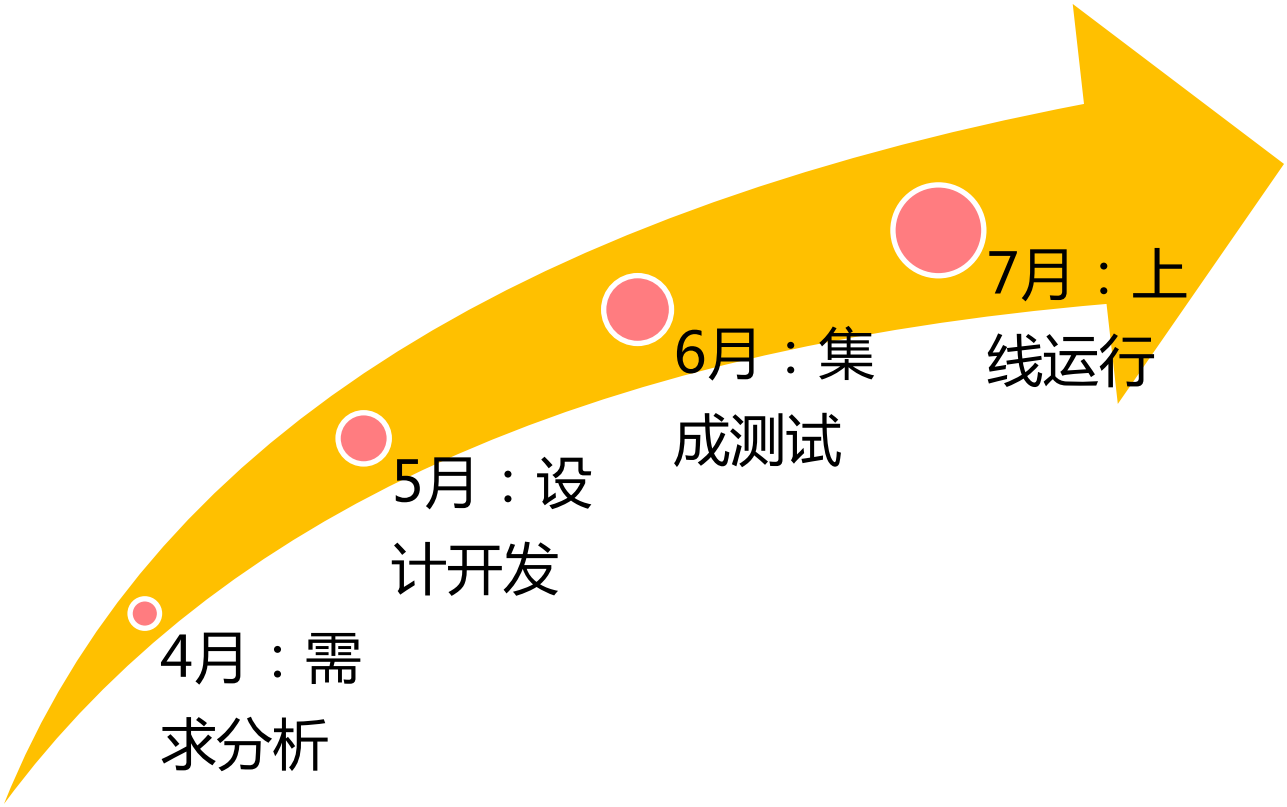
```
#!/bin/bash
#$1 jar
bash sparksubmit.sh com.aisino.bi.dlt.Dltfp $1
bash sparksubmit.sh com.aisino.bi.etl.fp.ZzsfpJx $1
bash sparksubmit.sh com.aisino.bi.etl.fp.JdcfpJx $1
bash sparksubmit.sh com.aisino.bi.etl.fp.HyfpJx $1
bash sparksubmit.sh com.aisino.bi.dlt.DltJxfpQd $1
bash sparksubmit.sh com.aisino.bi.etl.fp.ZzsfpJxQd $1
bash sparksubmit.sh com.aisino.bi.etl.fp.JdcfpJxQd $1
bash sparksubmit.sh com.aisino.bi.etl.fp.HyzpJxQd $1
```

TiD2016
质量竞争力大会
软件研发顶级盛会

下一代 SOFTWARE DEVELOPMENT 软件研发
更高 更新 更深

改进的效果

快速推进占领市场



开展深度分析与挖掘

TiD2016

企业精准画像

- 基于大数据分析，像绘制人脸一样绘制企业特征
- 更多企业特征分析
- 更详尽的企业分类

- 行业龙头企业
- 供应链核心企业
- 外贸型企业
- 快速增长型企业
- 稳定型企业
- 周期性企业

企业成长性分析

- 通过建模对每个企业未来一段时间的成长性进行预测
- 然后根据企业所在的地域
- 对各地区未来一段时间整体的发展状况进行预测
- 绘制出各地域未来发展状况分布统计图

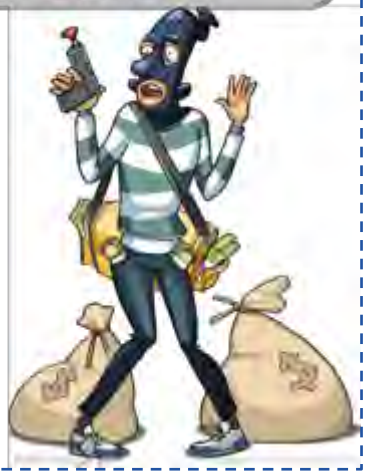
行业发展指数分析

- 根据增值税发票所体现的企业经营状况
- 结合企业所属的行业
- 分析各行业在各地域的平均发展水平
- 分析快速发展的行业
- 分析不景气的行业
- 为税收政策的制订提供数据支持

企业走逃预警监控

- 通过对以往走逃企业的大数据分析
- 整理并分析走逃企业的数据特征
- 从而成功预测将来可能走逃的企业

- 通过该监控进行重点企业监控
- 对走逃企业提前采取措施
- 同样的方法还可以监控骗税、虚开发票的企业



我的书...

明白真正的**专业级**软件开发是如何进行的

明白真正的**重构具体**是一步步怎么做的

高效可行的重构七步。面对实际重构，不会卡壳。

超越代码级重构，渗透系统与设计的各个层面。

第一次真正理解那些最熟悉的陌生技术，

惊讶于它们各就各位榫卯成强韧的整体。

范钢
微博：诗意联盟
微信：shiyilianmeng0
博客：http://fangang.iteye.com
http://www.cnblogs.com/mooodo