

“游戏开发与运维” 专场

出品人：涂彦

讲师：管鹤鸣、何少鹏、刘超

QCon

全球软件开发大会

10月17-19日 上海·宝华万豪酒店



扫码锁定席位

九折即将结束

团购还享更多优惠，折扣有效期至9月17日

扫描右方二维码即可查看大会信息及购票



如果在使用过程中遇到任何问题，可联系大会主办方，欢迎咨询！

微信：qcon-0410

电话：010-84782011

ArchSummit

全球架构师峰会 2017



扫码锁定席位

12月8-9日 北京·国际会议中心

七折即将截止立省2040元

使用限时优惠码AS200，

以目前最优惠价格报名ArchSummit

仅限前20名用户，优惠码有效期至9月19日，

扫描右方二维码即可使用



如果在使用过程中遇到任何问题，可联系大会主办方，欢迎咨询！

微信：aschina666

电话：15201647919

极客搜索

全站干货，一键触达，只为技术

s.geekbang.org



扫描二维码立即体验

有没有一种搜索方式，能整合 InfoQ 中文站、极客邦科技旗下12大微信公众号矩阵的全部资源？

极客搜索，这款针对极客邦科技全站内容资源的轻量级搜索引擎，做到了！

扫描上方二维码，极客搜索！

这里只有 技术领导者

EGO会员第二季招募季正式开启



E小欧

报名时间：9月1日-9月15日

扫描添加E小欧，
邀您进入EGO会员预报名群

立即报名



TABLE OF CONTENTS

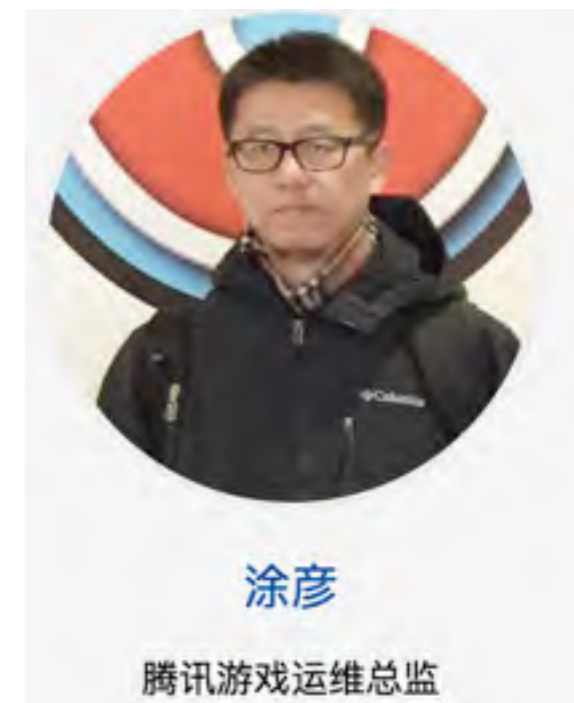
导读：

个人介绍

运维在游戏行业的机遇与挑战

AIOPS在游戏行业的思考

- 腾讯游戏运维
- 10年+，网络游戏行业
- 关注：运维在游戏行业的长期生存与发展
 - 基础游戏运维服务
 - 增值游戏运维服务



- 运维在游戏行业的机遇与挑战：全球化

- 代理

- 自研

- 海外



- AIOPS在游戏行业的思考：场景化
 - 基础运维：发布变更、故障管理、用户体验。。。。
 - 增值运维：技术运营。。。。

- 本场分享：
- 腾讯游戏：“运维+”，腾讯游戏海量运营的催化剂
- 盛大游戏：盛大游戏的深度网络优化实践
- 网易游戏：游戏基于容器的持续集成与简化运维深度实践

- TIPS

- 每场分享结束，5分钟提问，10分钟休息
- 会后任何问题，都可以去“讲师见面会”继续
- 离开时，门口工作人员有IPAD，记得给讲师投票，谢谢

"运维+", 腾讯游戏海量运营的 催化剂

管鹤鸣

腾讯游戏运维专家

TABLE OF CONTENTS

个人介绍

前言

开发+运维：合作共赢实现 $1+1>2$

开发+运维：我们都是游戏架构设计者

产品+运维：放开那个开发，让我来！



- 腾讯游戏运维专家
- 10年行业经验
- 多款端手游运维经验，涵盖从立项到公测整个业务完整生命周期

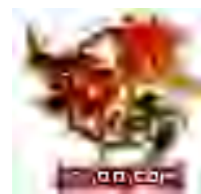


TABLE OF CONTENTS

个人介绍

前言

开发+运维：合作共赢实现 $1+1>2$

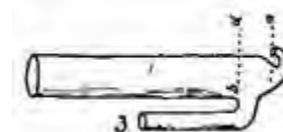
开发+运维：我们都是游戏架构设计者

产品+运维：放开那个开发，让我来！

魔术“神杯”的故事



瑞典化学家
琼斯·雅可比·贝采里乌斯
1779年8月20日-1848年8月7日



内容总览

您将了解到

对于腾讯游戏的海量运维管理，运维和游戏开发是如何一起打造零门槛进程集中管理系统的

从游戏运维角度出发，我们可以如何影响游戏的前后台架构，帮助提升可靠性和用户体验的

海量用户需要靠精细化运营来保障，腾讯游戏运维可以通过“蓝鲸”帮助产品运营提供哪些运营服务方案

TABLE OF CONTENTS

个人介绍

前言

开发+运维：合作共赢实现1+1>2

开发+运维：我们都是游戏架构设计者

产品+运维：放开那个开发，让我来！

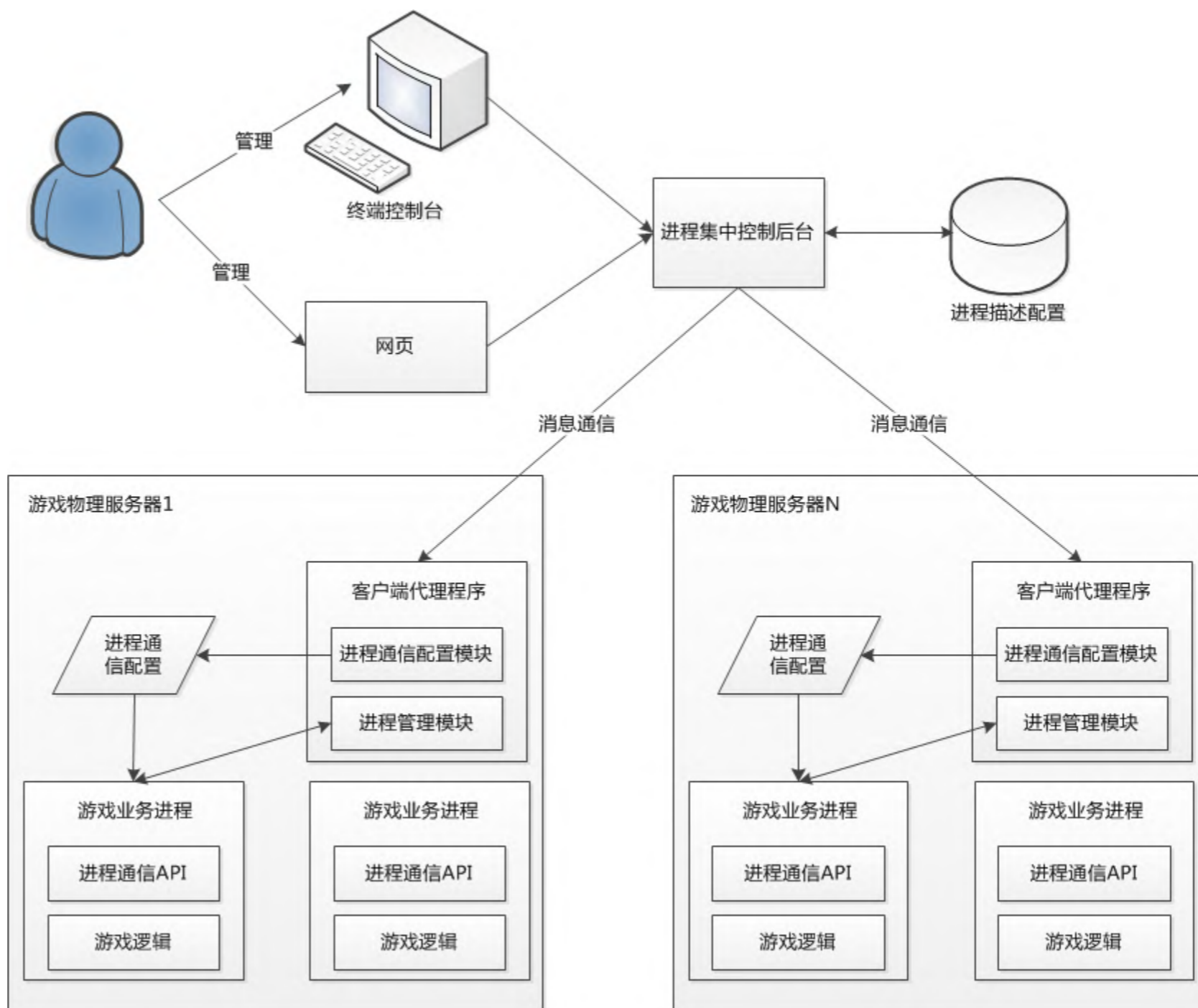
开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

进程集中管理系统

主要功能点

- 业务进程部署信息集中管理
- 业务进程启停，状态检查，自动拉起等控制
- 业务进程配置文件自动生成和传送
- 业务进程通信关系的自动生成和更新
- 数据收集与告警
- 支持命令行和console
- 支持作业流任务



开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

第一步：进程描述信息及层级关系

```
<TCMCENTER>
  <!--
  2)业务系统进程描述:请在以下部分描述业务系统进程
  -->
  <cluster workPath="/data/michaeldeng/user00/apps/tcm/example/tsdm" AutoTimeGap="10" OpTimeout="10" MsgRoundTime="10">
    <Proc FuncName="vimonster1" FuncID="101" Flag="Notapp" IsCommon="1"/>
    <Proc FuncName="vimonster2" FuncID="102" Flag="Notapp" IsCommon="1"/>
    <Proc FuncName="vimonster3" FuncID="103" Flag="Notapp" IsCommon="1"/>
    <world Isolated="1" OpTimeout="30" MsgRoundTime="20">
      <Proc FuncName="tbusd" FuncID="100" Seq="1" WorkPath="tbusd/bin" Agrs="--conf-file ../cfg/tbusd.xml"/>

      <!--公共接入进程,多zone共用-->
      <Proc FuncName="tconnd_common" ProcName="tconnd" FuncID="11" Seq="21" WorkPath="tconnd_common/bin" Agrs="--conf-file
$instance_log.xml"/>
      <Proc FuncName="operd_common" ProcName="operd" FuncID="12" Seq="20" WorkPath="operd_common/bin" Agrs="--conf-file
$instance_log.xml"/>
      <zone Isolated="1" OpTimeout="10">

        <Proc FuncName="tconnd" FuncID="1" Seq="15" WorkPath="tconnd/bin" Agrs="--conf-file=../cfg/tconnd_${world}.${zone}
$instance_log.xml"/>
        <Proc FuncName="tconnd_1monster" ProcName="tconnd" FuncID="2" Seq="15" WorkPath="tconnd/bin" Agrs="--conf-file
$instance_log.xml"/>

        <Proc FuncName="operd" FuncID="200" Seq="14" WorkPath="operd/bin" Agrs="--conf-file=../cfg/operd_${world}.${zone}
$instance_log.xml"/>
        <Proc FuncName="operd_1monster" ProcName="operd" FuncID="201" Seq="14" WorkPath="operd/bin" Agrs="--conf-file
$instance_log.xml"/>

        <Proc FuncName="transd" FuncID="210" Seq="13" WorkPath="transd/bin" Agrs="--conf-file=../cfg/transd_${world}.${zone}
$instance_log.xml"/>
        <Proc FuncName="dmsd" FuncID="220" Seq="12" WorkPath="dmsd/bin" Agrs="--nthread=8 --conf-file=../cfg/dmsd_${world}.${zone}
$instance_log.xml"/>
      </zone>
    </world>
  </cluster>
```

开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

第二步：对进程根据功能定义进程组

```
<!--  
3)业务系统进程组定义:请在以下部分将业务系统进程根据功能定义成不同的进程组  
请确保进程组中的进程在<业务系统进程描述>部分中定义  
-->  
  
<!--公用接入组-->  
<ProcGroup Name="connect_common" Layer="world">  
  <Proc FuncName="operd_common" />  
  <Proc FuncName="tconnd_common" />  
</ProcGroup>  
  
<!--专属接入组-->  
<ProcGroup Name="connect" Layer="zone">  
  <Proc FuncName="operd" />  
  <Proc FuncName="tconnd" />  
</ProcGroup>  
  
<!--事务组-->  
<ProcGroup Name="trans" Layer="zone">  
  <Proc FuncName="transd" />  
</ProcGroup>  
  
<!--数据服务组-->  
<ProcGroup Name="dms" Layer="zone">  
  <Proc FuncName="dmsd" />  
</ProcGroup>  
  
<!--bus中转-->  
<ProcGroup Name="tbusd" Layer="world">  
  <Proc FuncName="tbusd" />  
</ProcGroup>  
  
<!--imonster专属接入组-->  
<ProcGroup Name="operd_imonster" Layer="zone">  
  <Proc FuncName="operd_imonster" />  
</ProcGroup>
```

开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

第三步：描述主机信息

```
tcmcenter>
<!--
4)业务系统主机描述:请将业务系统用到的主机在下面进行描述,IsVirtual默认为0, IsVirtual=1表示该主机为虚拟主机
-->
<HostTab >
  <Host Name="Host1" InnerIP="172.25.40.208" >
  </Host>

  <Host Name="vHostImonster1" IsVirtual="1" InnerIP="172.25.40.101" >
  </Host>

  <Host Name="vHostImonster2" IsVirtual="1" InnerIP="172.25.40.102" >
  </Host>

  <Host Name="vHostImonster3" IsVirtual="1" InnerIP="172.25.40.103" >
  </Host>

</HostTab>
/tcmcenter>
```

开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

第四步：描述进程组在物理主机上的分布

```
<tcmlcenter>
<!--
5)业务系统进程组部署:请在以下部分描述各进程组在各主机上的部署情况
请确保进程组和主机都已经在相应的各部分定义
-->
<ClusterDeploy CustomAttr="">
  <DeIoyGroup Group="VImonster2" Host="VHostImonster2" InstID="1" CustomAttr=""/>
  <DeIoyGroup Group="VImonster1" Host="VHostImonster1" InstID="1" CustomAttr=""/>

  <world ID="1" CustomAttr="test200">
    <DeIoyGroup Group="tbusd" Host="Host1" InstID="1" CustomAttr=""/>
    <!--部署公共接入进程组-->
    <DeIoyGroup Group="VImonster2" Host="VHostImonster2" InstID="1" CustomAttr=""/>
    <zone ID="1" CustomAttr="">
      <DeIoyGroup Group="operd_imonster" Host="Host1" InstID="1" CustomAttr=""/>
      <DeIoyGroup Group="tconnd_imonster" Host="Host1" InstID="1" CustomAttr=""/>
      <DeIoyGroup Group="trans" Host="Host1" InstID="1" CustomAttr=""/>
      <DeIoyGroup Group="dms" Host="Host1" InstID="1" CustomAttr=""/>
      <DeIoyGroup Group="VImonster3" Host="VHostImonster3" InstID="1" CustomAttr=""/>
    </zone>
  </world>
</ClusterDeploy>
</tcmlcenter>
```

开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

第五步：描述各进程通信关系

```
<!--
6)业务系统进程间bus通信关系:请在以下部分描述各进程tbus通信关系
请确保进程已经在进程描述部分定义
-->
<TbusConf BussinessID="0" GCIMShmKey="5600" GRMShmKey="7800" TbusdServicePort="9021" channelSize="204800" TbusdConfPath="/

  <!--公用tonnd_common和operd_common一一对应-->
  <Channel ProcSet1="tconnd_common" ProcSet2="operd_common" Mask="255.255.0.255" />

  <!--公用operd_common可以访问所有的transd-->
  <Channel ProcSet1="operd_common" ProcSet2="transd" Mask="255.0.0.0" />

  <!--业务专属operd和tconnd一一对应-->
  <Channel ProcSet1="tconnd" ProcSet2="operd" Mask="255.255.0.255" />

  <!--iMonster业务专属operd和tconnd一一对应-->
  <Channel ProcSet1="tconnd_imonster" ProcSet2="operd_imonster" Mask="0.0.0.0" />

  <!--iMonster业务接入进程-->
  <Channel ProcSet1="vimonster1" ProcSet2="operd_imonster" Mask="0.0.0.0" />
  <Channel ProcSet1="vimonster2" ProcSet2="operd_imonster" Mask="0.0.0.0" />
  <Channel ProcSet1="vimonster3" ProcSet2="operd_imonster" Mask="0.0.0.0" />

  <!--业务专属的operd可以访问业务内部的transd-->
  <Channel ProcSet1="operd_imonster" ProcSet2="transd" Mask="255.255.0.0" />

  <!--业务的transd可以访问同业务所有dmsd-->
  <Channel ProcSet1="transd" ProcSet2="dmsd" Mask="255.255.0.0" />
</TbusConf>
</tcmcenter>
```

开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

实际使用中的问题



影响

部署易出错
沟通成本巨大
排查和调试时间长

开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

一起来优化

我了解进程描述

我了解生产环境

那我定义该角色
上跑什么应用

我告知服务器
属于什么角色

我们做成开关项
同时给你自定义
增加应用的功能

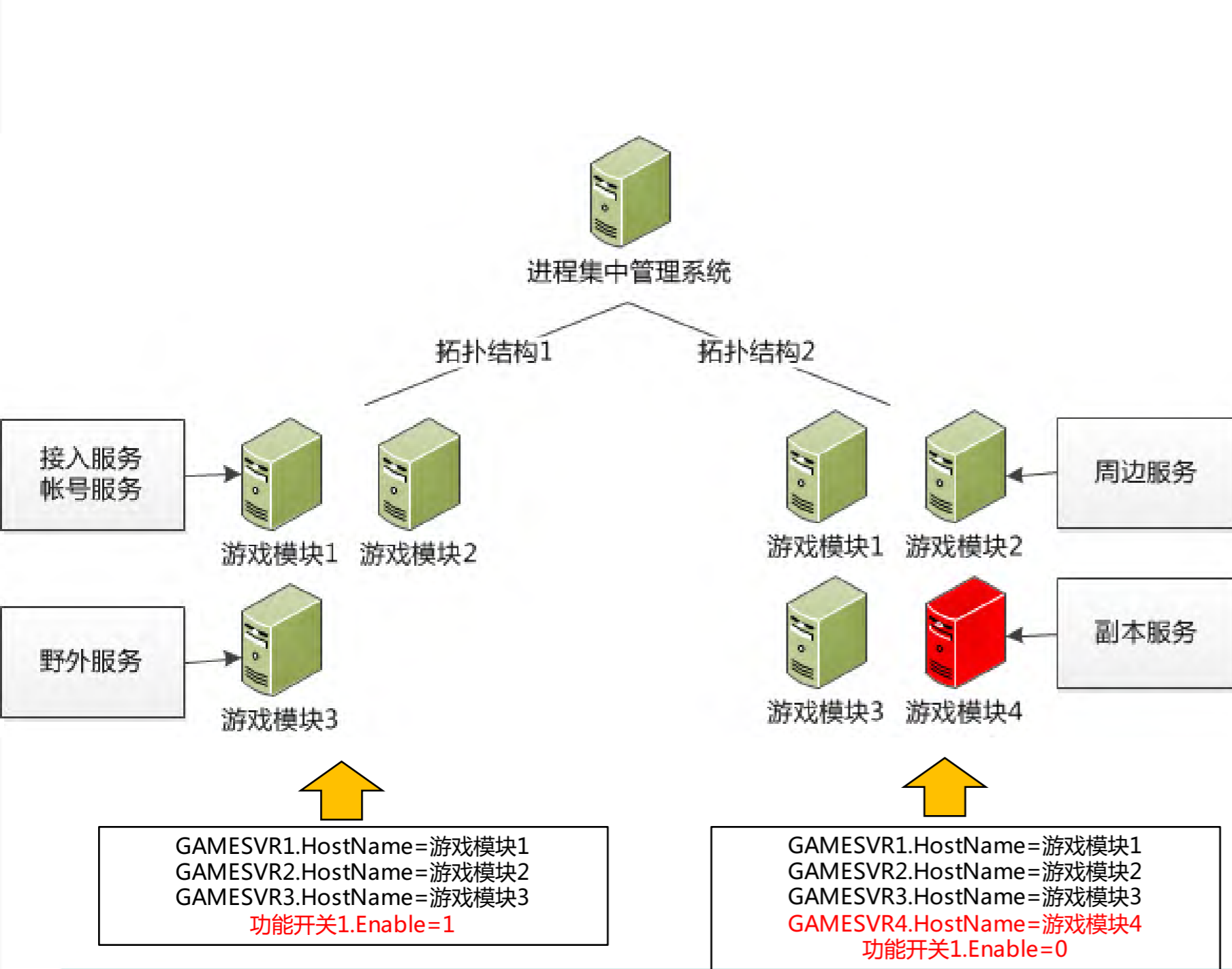
有时我需要开关
功能和部署其他
额外周边应用



合作共赢

开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

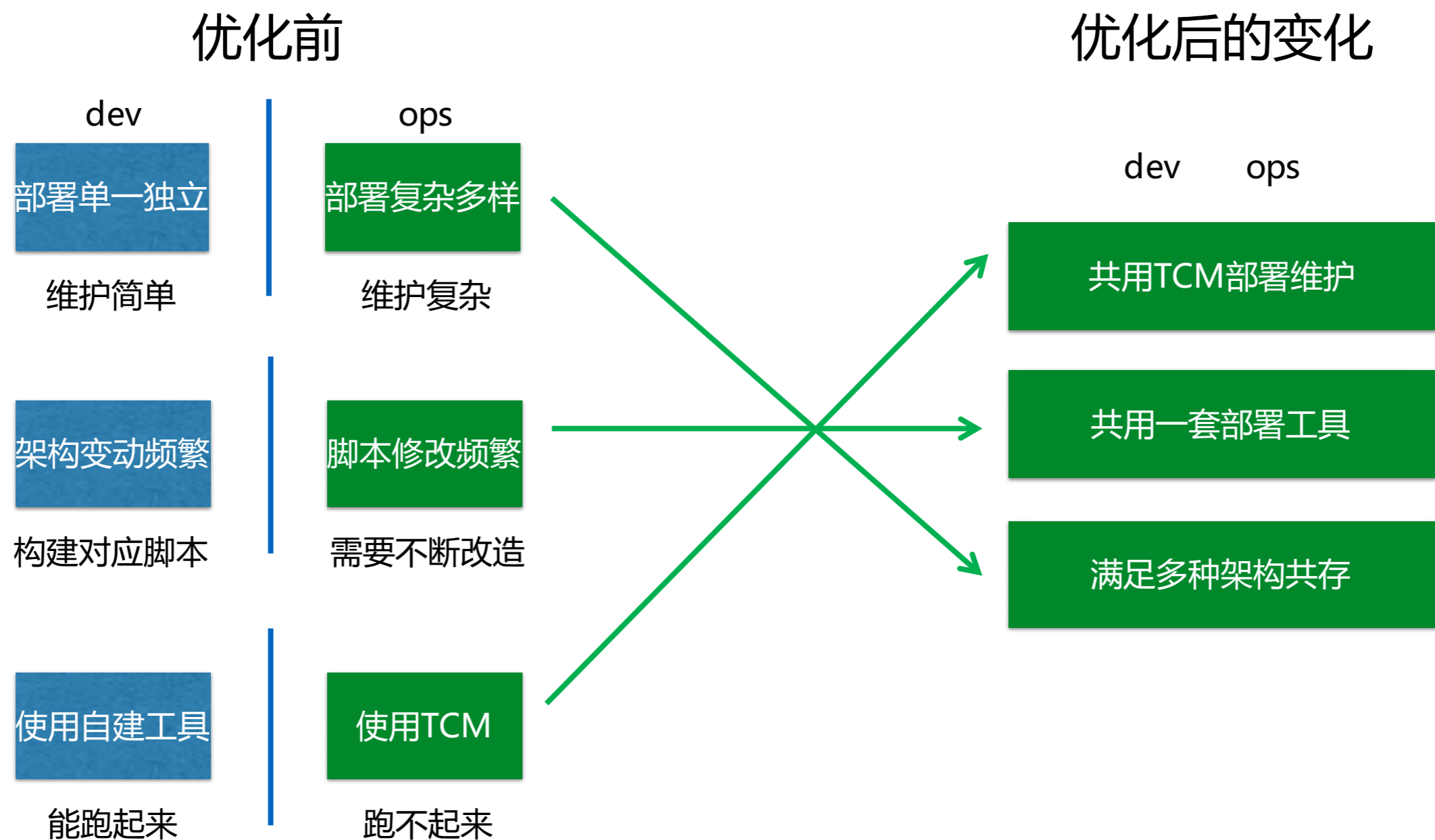


```
微端不删档801
C:\WORLD_8_01
GAMESVR1.HostName = 801tgs
GAMESVR2.HostName = 801ttc
GAMESVR_TCDHND.EnableTgw = 1
GAMESVR_TCDHND.MaxConnection = 6200
GAMESVR_TCDHND.PermitLogin = 6000
GAMESVR_TCDHND.UseVasKey = 1
GAMESVR_SCENE.EnableEnter = 1
GAMESVR_LDG.HostName = 801tlog
GAMESVR_DB.HostName = 801gdb
GAMESVR_DB.DBNamePostfix = sh801
GAMESVR_DB.DBUsername = [REDACTED]
GAMESVR_DB.DBPassword = [REDACTED]
GAMESVR_DRDB.HostName = 801gdr
GTS.EnableGtsc = 0
GTS.TargetWorldID = 0
AAS.EnableAntiAddiction = 1
ANTIBOT.Enable = 0
TWATCH.Enable = 0
GAMESVR.EnableGfgServer = 0
GAMESVR.EnableIQGroup = 1
GAMESVR_DSSDB.DBNamePostfix = 801
GAMESVR.CrossEGRWorldId = 0
GAMESVR.EnableCrossIn = 0
GAMESVR.EnableCrossOut = 0
GAMESVR.CrossIdcId = 8
```

```
[SHELL sa_playeroff]
FuncID = 70
FuncName = sa_playeroff
ProcName = prepare_stopping.sh
WorkPath = world/bin/
ConfigPath = ./
Flag = runshell
IsCommon = 1
DeployHost = ALL_WORLD_GSSVR
```

开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统



开发+运维：合作共赢实现1+1>2

打造基于拓扑结构的零门槛进程集中管理系统

某游戏业务优化效果：

- 游戏服务器逐年递增的情况下 ↑
- 部署错误率从41.67%降低至0% ↓
- 平均部署时长缩短80% ↓

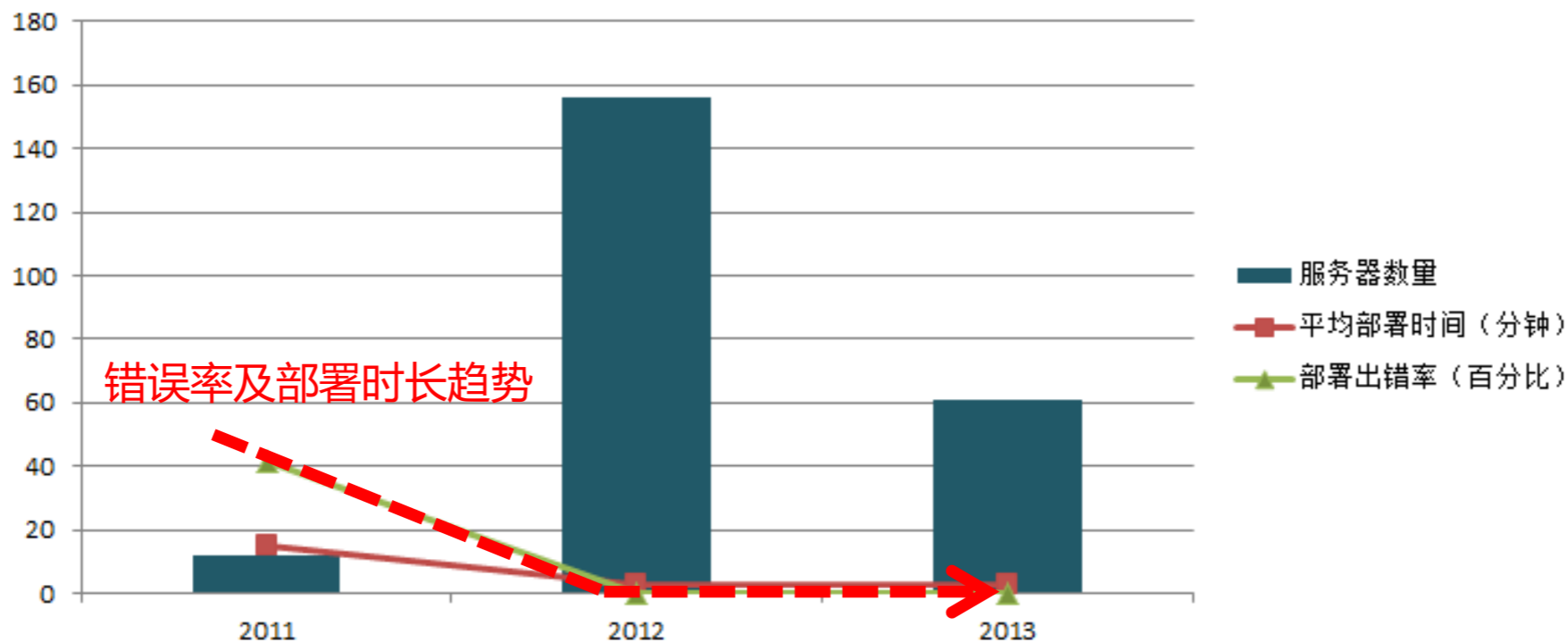


TABLE OF CONTENTS

个人介绍

前言

开发+运维：合作共赢实现1+1>2

开发+运维：我们都是游戏架构设计者

产品+运维：放开那个开发，让我来！

开发+运维：我们都是游戏架构设计者



看到的风景
取决于
所在的角度



开发+运维：我们都是游戏架构设计者

某MMORPG业务后台架构



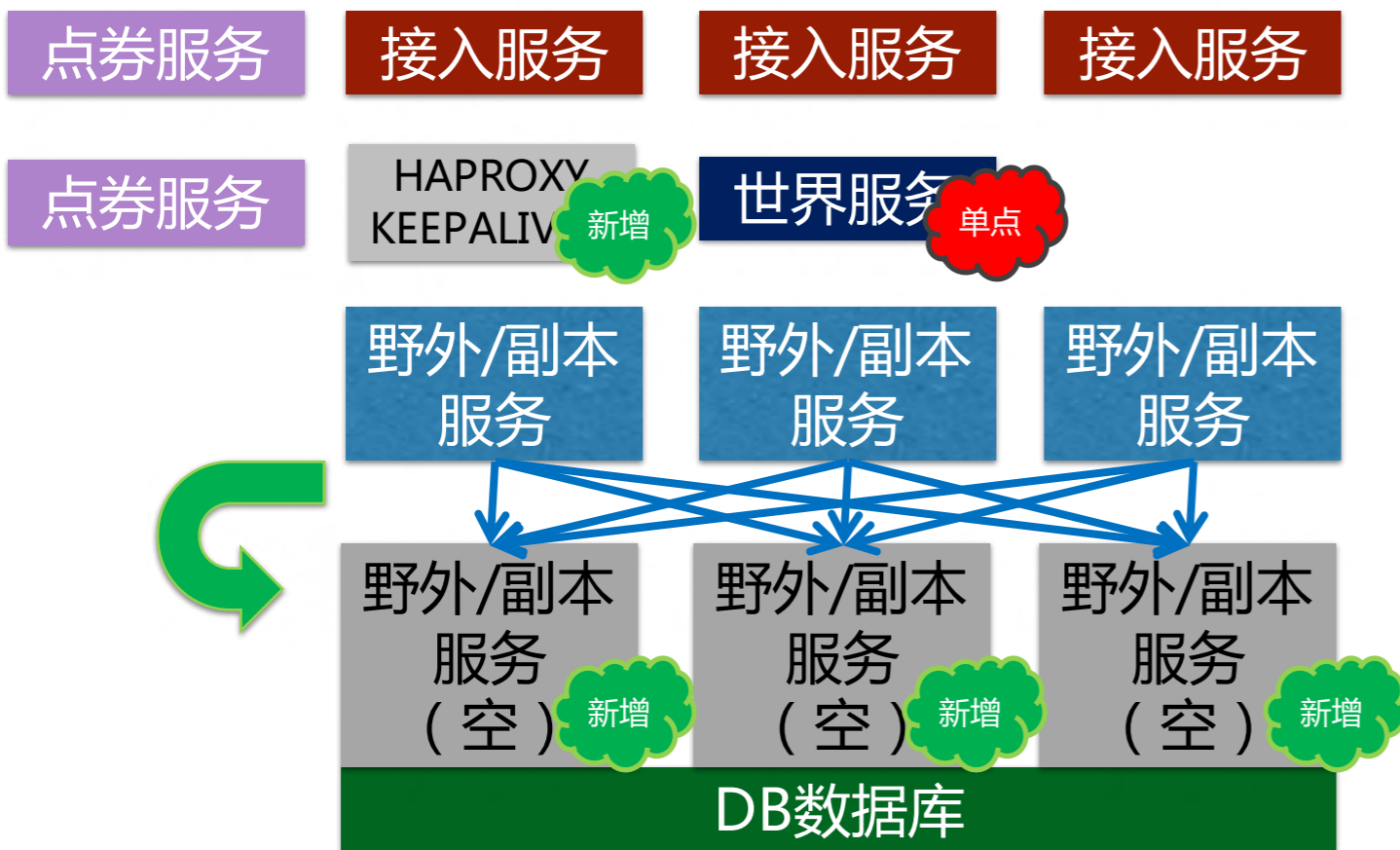
架构缺点：

- ◆ 任何单点故障都需要停机维护
- ◆ 点券服务的主备模式存在性能瓶颈
- ◆ 运营后开新服帐号一致性
- ◆ 合服逻辑复杂，例如角色名

开发+运维：我们都是游戏架构设计者

某MMORPG业务后台架构优化后

单服级别



IDC级别

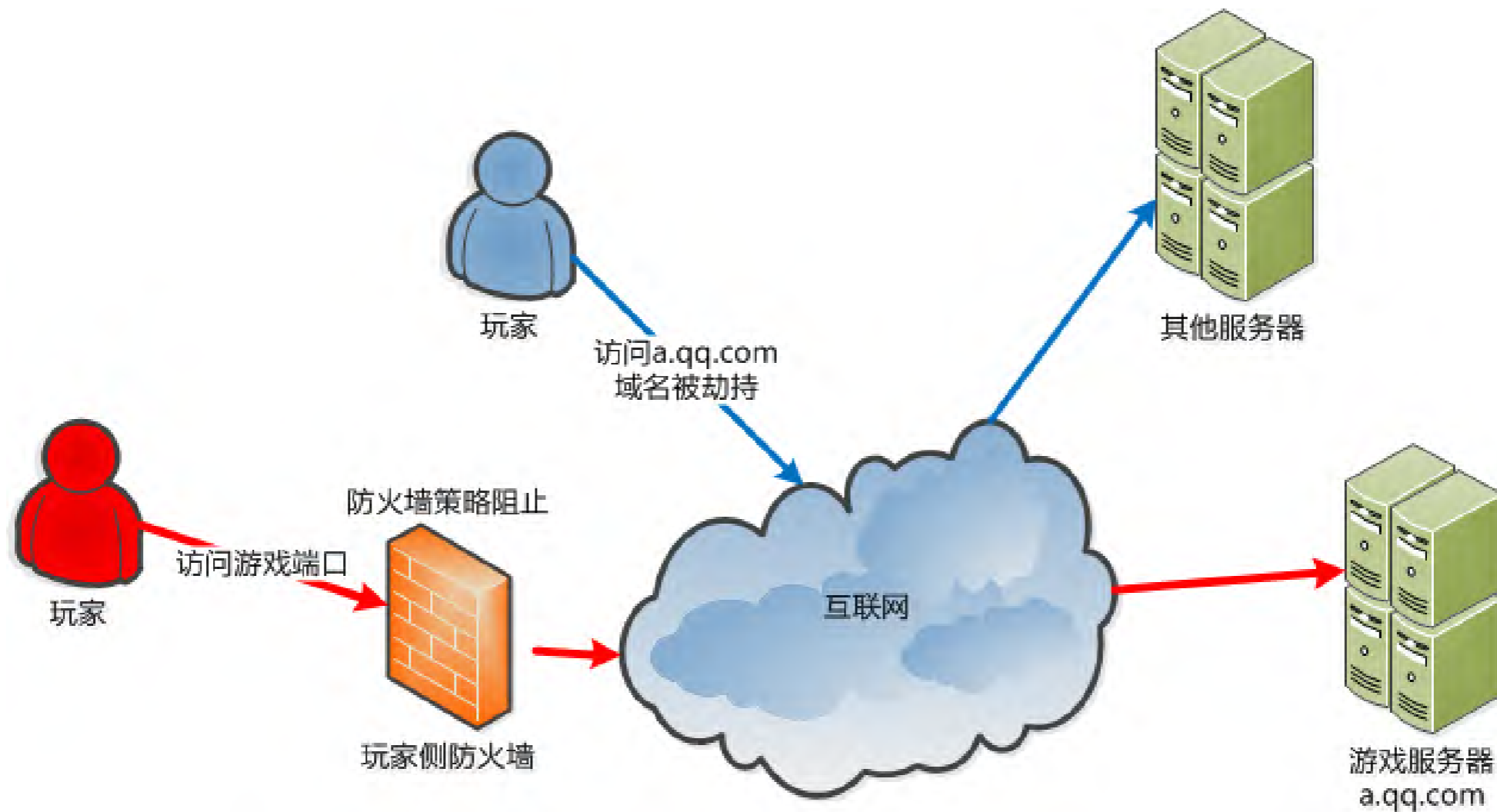


优化收益：

- ◆ 副本漂移技术解决了部分的单点
- ◆ 点券服务性能瓶颈解决
- ◆ 解决帐号一致性问题
- ◆ 名字服务改为IDC级别简化合服逻辑

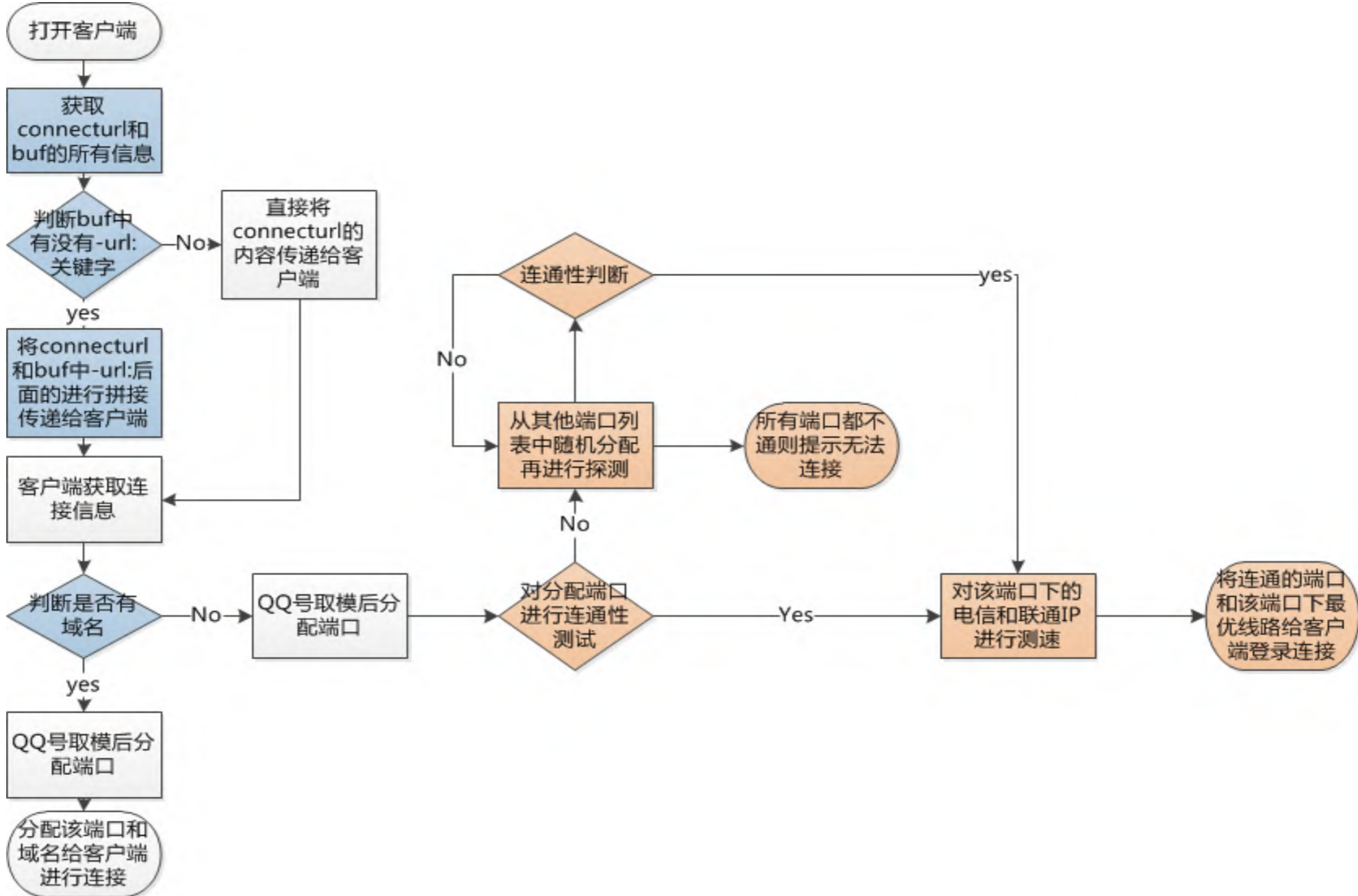
开发+运维：我们都是游戏架构设计者

某MMORPG业务客户端登录优化



开发+运维：我们都是游戏架构设计者

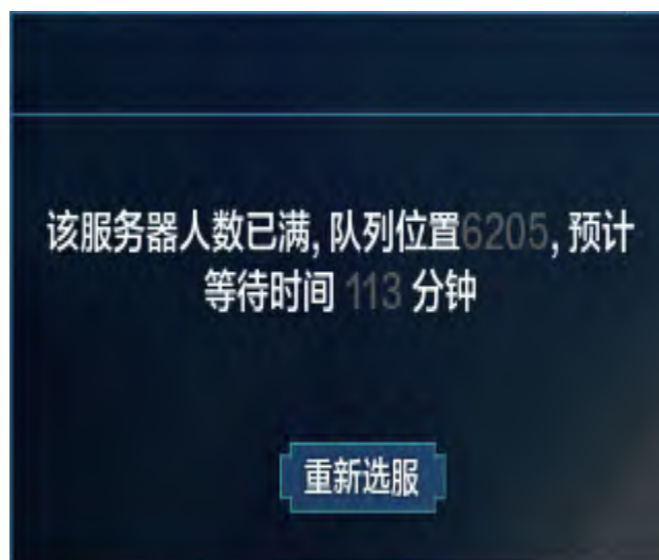
某MMORPG业务客户端登录优化



开发+运维：我们都是游戏架构设计者

某MMORPG业务客户端登录优化

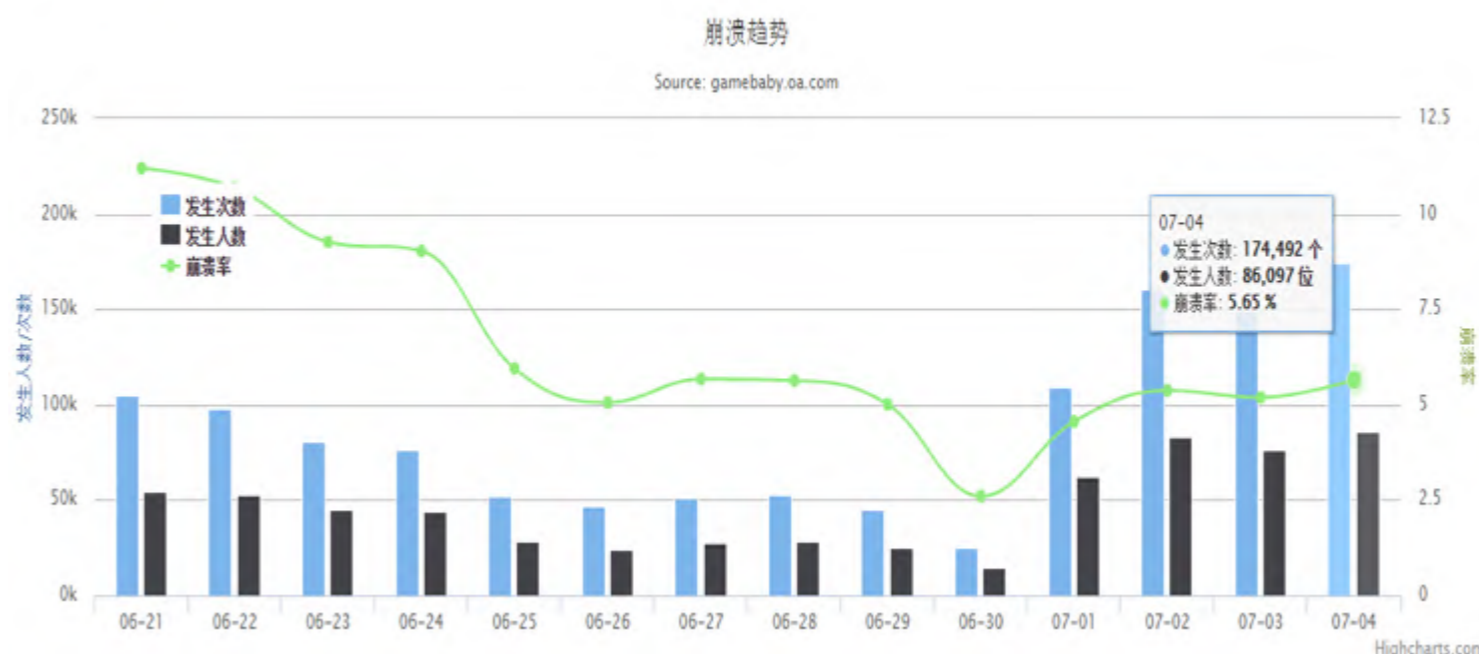
当你玩游戏到一半，客户端闪退
当你排队排了几小时，重新打开后
看到队列位置的数字后，你还愿意等吗？
(我们的一个新服排队最高3W)



每天有多少次客户端意外崩溃？
约17W次，8.6W人。
(还没包括排队中掉线的)

他们如果因为排队而流失...

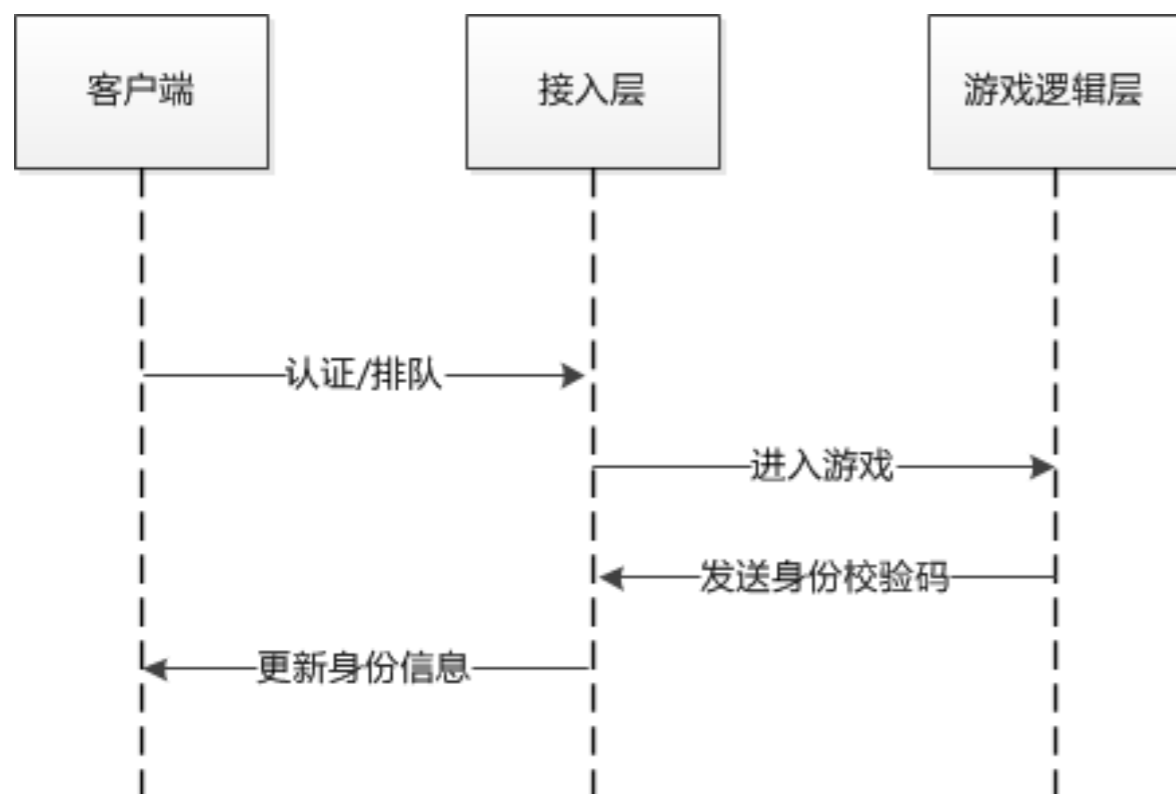
崩溃走势(2week)



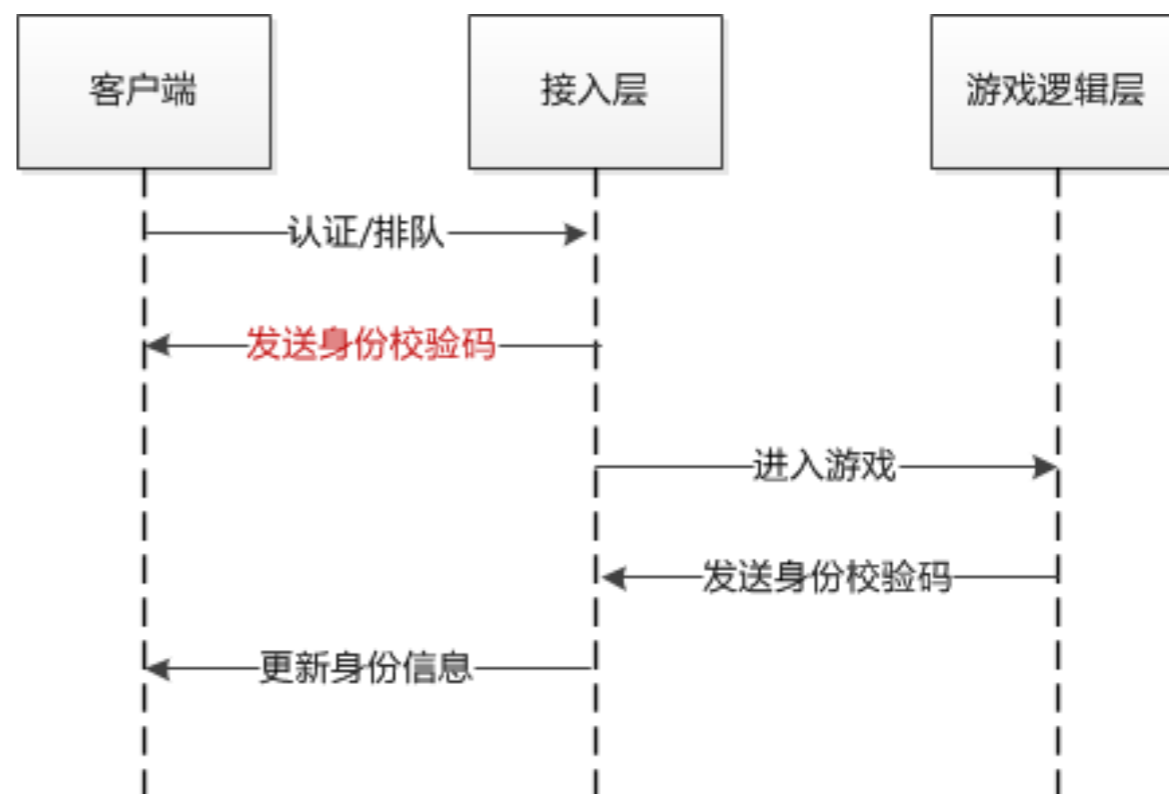
开发+运维：我们都是游戏架构设计者

某MMORPG业务客户端登录优化

断线重连逻辑优化前



断线重连逻辑优化后



解决有排队时的掉线保护问题

开发+运维：我们都是游戏架构设计者

某MMORPG业务客户端登录优化

客户端闪退掉线保护原理

优化前：客户端登录游戏后获取服务器下发的登录凭证，并存在内存中

缺点：客户端崩溃后，内存数据丢失，凭证失效

优化后：正常登录后客户端把获取到的登录凭证进行加密，落地到本地，并设置5分钟有效时长，到点和正常退出后销毁

TABLE OF CONTENTS

个人介绍

前言

开发+运维：合作共赢实现 $1+1>2$

开发+运维：我们都是游戏架构设计者

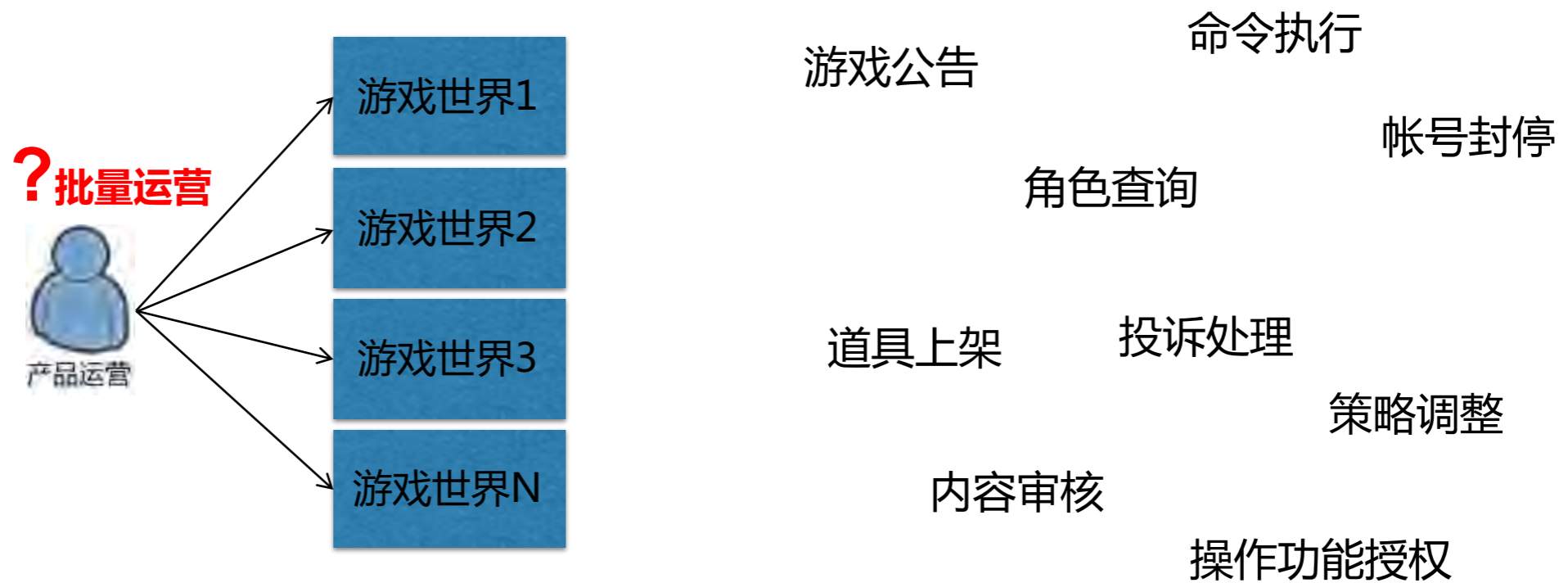
产品+运维：放开那个开发，让我来！

产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案

场景一：多数代理游戏开发商只提供单服管理工具，或根本没有，或功能不满足需求，海量运营时可能存在几十甚至上百游戏世界

如何破？



产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案

腾讯运维可为游戏业务提供定制化的一站式运营平台

公告管理

运营操作 > 公告作业管理

模板管理

添加模板

公告1
请勿相信游戏内的各种虚假礼包领奖信息页面，以防账号被盗，造成...

公告2
官方不会以任何形式向玩家索要游戏账号与密码，请广大玩家们时刻...

活动引导
【活动】收集图鉴能获得! 图鉴、图鉴、图鉴! 参与集章...

公告任务

大区选择: × 9114_ × 9120_ × 9102_ 批量大区选择

公告模板: 公告1 轮播方式: 每日 轮播间隔(秒): 60


开始日期: 2017-09-05 结束日期: 2017-09-12 开始时间: 20:23:50 结束时间: 20:23:50

提交公告任务 取消

产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案

内容审核

未审核		已审核		全部		
大区	QQ号	角色ID	昵称	时间	图片	操作
1001	[REDACTED]	26014[REDACTED]59415	天[REDACTED]瓜	2017-09-05 19:14:03		<input type="button" value="通过"/> <input type="button" value="下架"/>
1001	46[REDACTED]4	260140[REDACTED]8217	明[REDACTED]包	2017-09-05 19:13:54		<input type="button" value="通过"/> <input type="button" value="下架"/>
1001	53[REDACTED]	26004[REDACTED]723987	砂[REDACTED]	2017-09-05 19:13:44		<input type="button" value="通过"/> <input type="button" value="下架"/>

产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案

场景二：产品运营在做运营活动时，如何选择在哪里做？活动期间用户的体验怎么样？

谁告诉我？

寻找GM活动 地图寻宝

野外刷怪 玩家聚集

决策参考 组织活动

产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案

用户分布热点地图



产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案

场景三：业务爆发期为了服务器的稳定，需要制定平滑放人的运营策略

什么时候放人？这么多服人工调？

你只有一条在线图



?手工放人



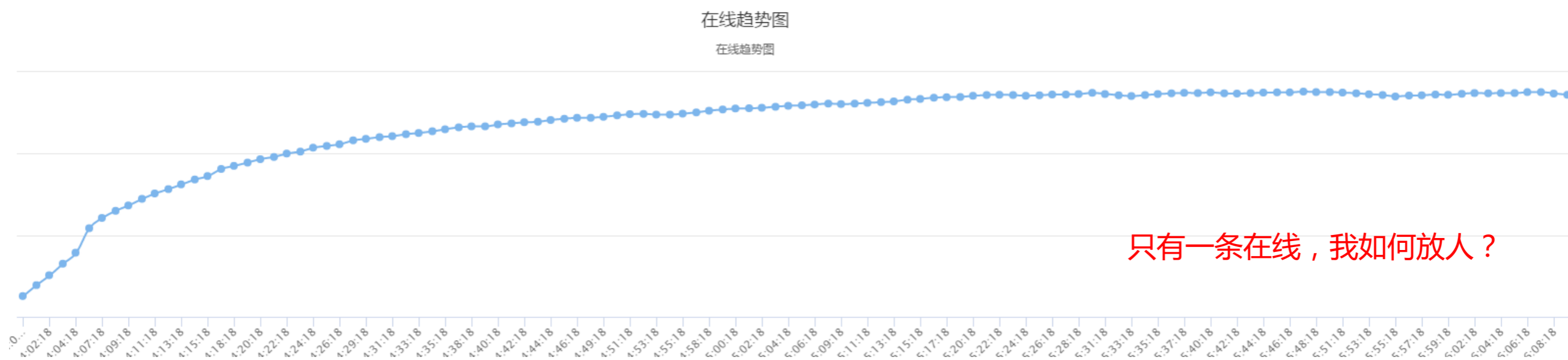
产品运营

游戏世界1	排队3000
游戏世界2	排队5000
游戏世界3	排队1860
游戏世界N	排队2630

产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案

放人决策因素之一：副本人数过高时不宜放人



产品+运维：放开那个开发，让我来！

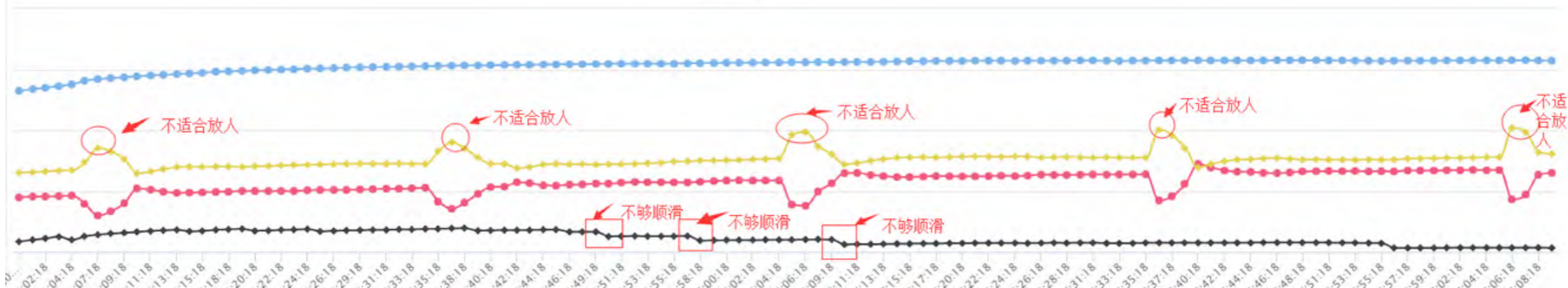
基于腾讯蓝鲸的一站式运营服务方案

精细化的在线视图



在线趋势图
在线趋势图

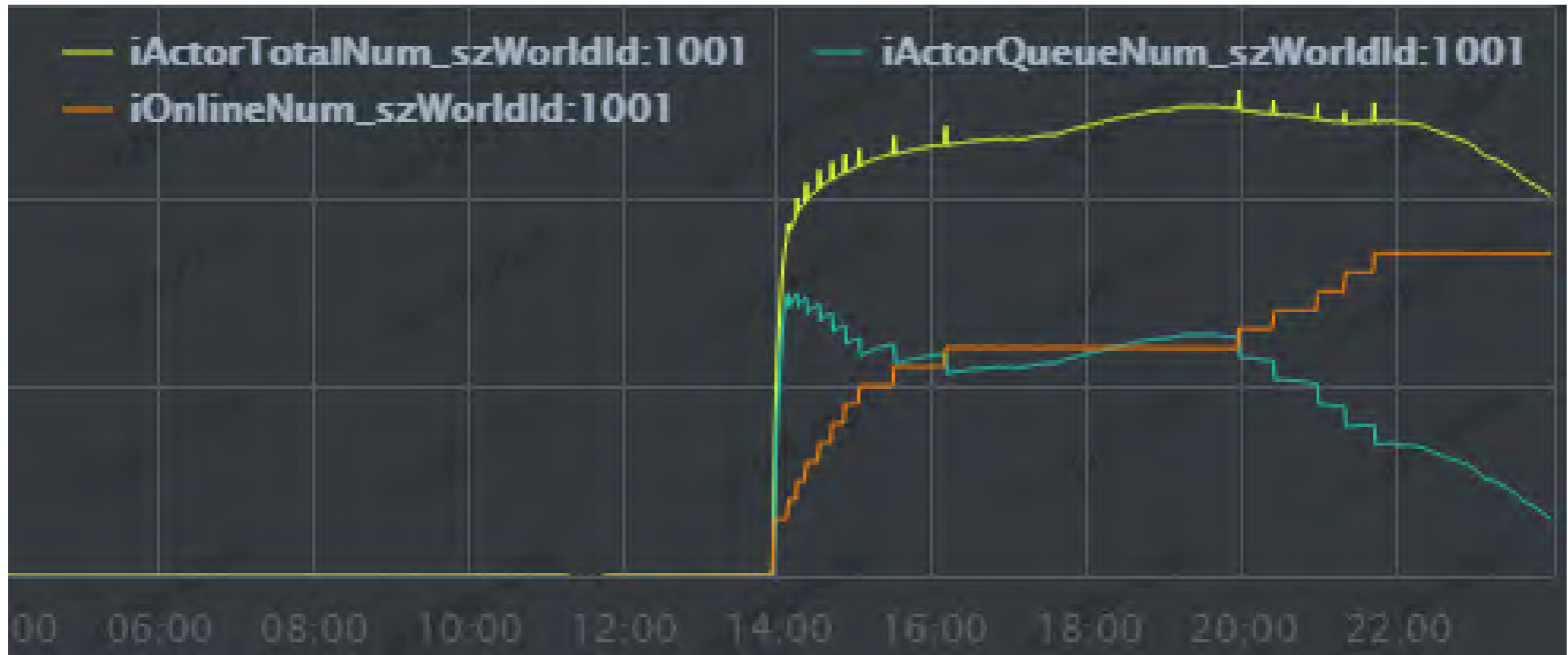
看到了冰山全景，放人的点就一览无遗



产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案

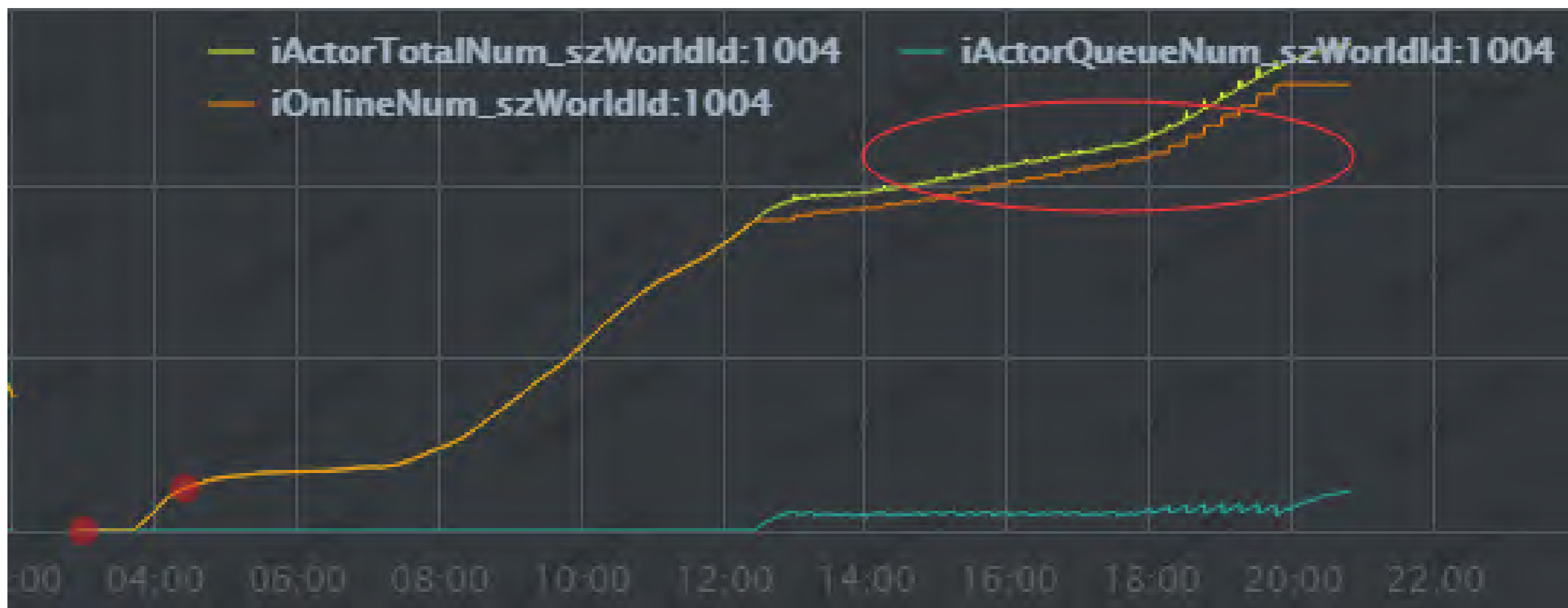
找到了合适的放人的时机，可是人工设置排队时并不平滑
(几十个服调到你崩溃)



产品+运维：放开那个开发，让我来！

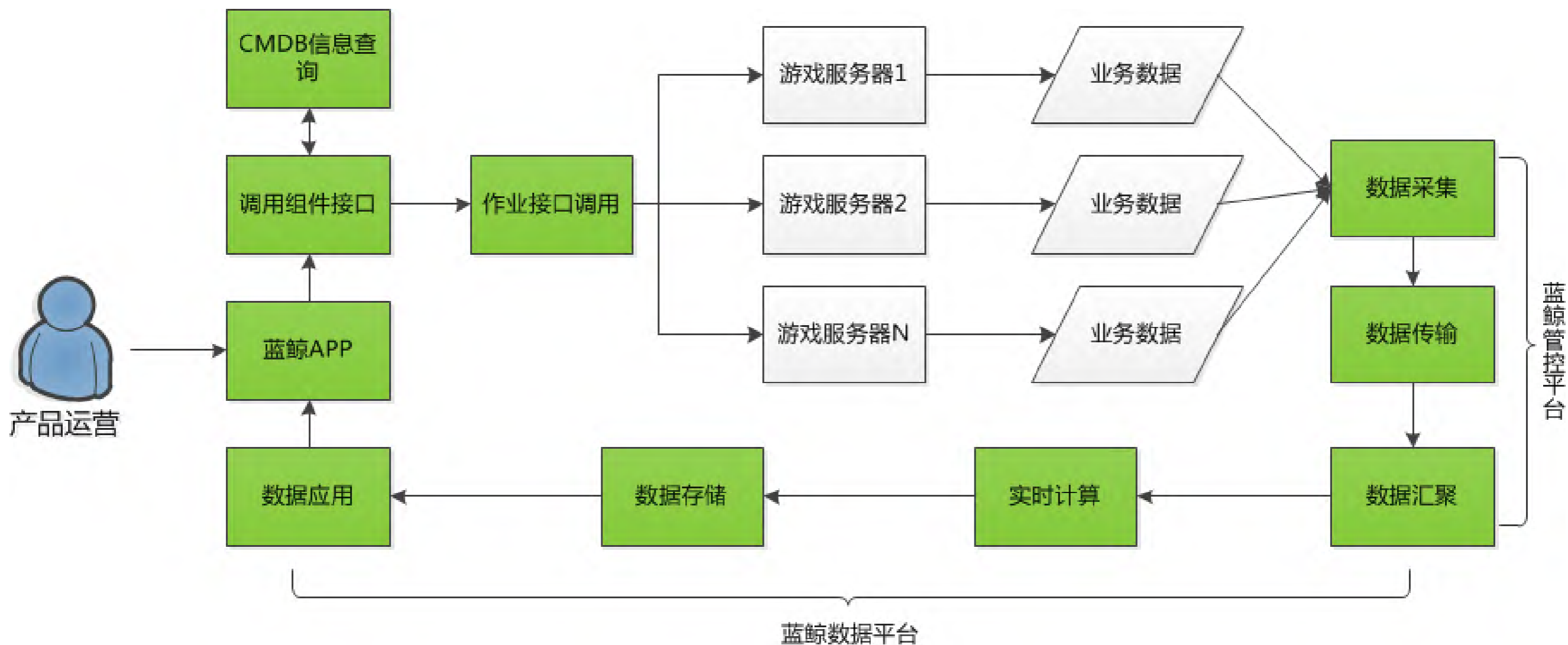
基于腾讯蓝鲸的一站式运营服务方案

根据放人维度智能调排队（无人值守）



产品+运维：放开那个开发，让我来！

基于腾讯蓝鲸的一站式运营服务方案



THANKS!

智能时代的新运维

CNUTCon 2017