3.模型训练和预测

本次实验是监督学习，因为我们已经知道每个样本是否患有心脏病，所谓监督学习就是已知结果来训练模型。解决的问题是预测一组用户是否患有心脏病。

1）拆分

首先通过拆分组件将数据分为两部分，本次实验按照训练集和预测集7：3的比例拆分。训练集数据流入逻辑回归二分类组件用来训练模型，预测集数据进入预测组件。

2）逻辑回归二分类

逻辑回归是一个线性模型，在这里通过计算结果的阈值实现分类。具体的算法详情推荐大家在网上或者书籍中自行了解。逻辑回归训练好的模型可以在模型页签中查看。

图6

3)预测

预测组件的两个输入分别是模型和预测集。预测结果展示的是预测数据、真实数据、每组数据不同结果的概率。

4.评估

通过混淆矩阵组件可以评估模型的准确率等参数：


混淆矩阵

混淆矩阵 比例矩阵 统计信息

模型	正确数	错误数	总计	正确率	准确率	召回率	F1指标
0	40	8	48	84.146%	83.333%	88.889%	86.022%
1	29	5	34	84.146%	85.294%	78.378%	81.69%

图7

通过此组件可以方便的通过预测的准确性来评估模型。

四.总结

通过以上数据探索的流程我们可以得到以下的结论。

1）特征权重

我们可以通过过滤式特征选择得到每个特征对于结果的权重。

featname	weight
thalach	0.16569171224597157
oldpeak	0.14640697618779352
thal	0.13769166559906015
ca	0.11467097546217575
chol	0.10267709576600859
age	0.07876430484527841
trestbps	0.0772599125640569
slop	0.07702762609078306
restecg	0.015246832497405105
cp	0.0037507283721422424
exang	0
fbs	0
sex	0

图8

-可以看出thalach(心跳数)对于是否发生心脏病影响最大。

-性别对于心脏病没有影响

2）模型效果

通过上文提供的14个特征，可以达到百分之八十多的心脏病预测准确率。模型可以用来做预测，辅助医生预防和治疗心脏病。

专题聚焦

教你玩转机器学习

专题简介：没有什么比大量新鲜生动的例子，更能帮助大家知道如何利用机器学习来解决实际生活中遇到的需求。玩转数据，让机器学习应用变得更加简单。

本主题链接：
https://yq.aliyun.com/topic/61

机器学习应用没那么难，这次教你玩心脏病预测
阿里云机器学习平台——PAI平台
人口普查统计案例
实践！如何用阿里云的机器学习得出泰坦尼克号沉船事件中谁有更大的概率获救
利用图算法实现金融行业风控
听说啤酒和尿布很配？本期教你用协同过滤做推荐
农业贷款发放预测
文本分析算法实现新闻自动分类
机器学习算法的离线调度实现-广告CTR预测

德歌：PostgreSQL独孤九式搞定物联网

摘要：随着物联网的越来越广泛使用，特定应用场景的需求也越发明显，如智能物流中需要对地理位置信息处理需求强烈；公安刑侦中的模糊化搜索等等。这不仅对物联网中的硬件是个挑战，同时对物联网中数据库管理系统也提出了更高的要求。分享关于PostgreSQL如何搞定物联网的“独孤九式”。

物联网行业不再仅仅只是设备的接入，设备接入后数据的采集和融合，以及融合后的分析，会为整个社会带来重要的价值。数据，让我们更真实的了解社会与自然，让人与自然、与社会更加的融合。但物联网也远没你想的那么难，经典的物联网架构分为感知层、网络层和应用层。感知层主要包括传感器网关、节点等数据采集工具；采集到的数据再经过互联网、移动通信网等传输网传递到物联网的“大脑”–应用层加以分析应用。随着物联网的越来越广泛使用，特定应用场景的需求也越发明显，如智能物流中需要对地理位置信息处理需求强烈；公安刑侦中的模糊化搜索等等。这不仅对物联网中的硬件是个挑战，同时对物联网中数据库管理系统也提出了更高的要求。本文即为大家分享关于PostgreSQL如何搞定物联网的“独孤九式”。

总诀式–知己知彼、百战不殆

图1: 总诀式–知己知彼、百战不殆

正如兵家讲究知己知彼，百战不殆一样，要真正实现万物互联、互通的物联网，就要熟知特定场景的具体要求，有针对性地给出解决方案。通过对智能家居、环境监测、城市交通、个人保健等具体场景的分析，可以对物联网应用场景特性做一个小结：

数据量大（压缩、数据处理能力）；

数据有时序、时空、文本属性（时序、地理位置、文本数据处理能力）；

某些数据难以结构化，如图像处理 （自定义能力、扩展能力、非结构化数据处理能力）；

数据处理实时性高（流式处理能力）；

数据维度多，相关性复杂（复杂查询、统计分析能力）；

有模糊、相似度查询需求（数据归类、索引功力）；

某些场景行锁竞争强烈（秒杀特性功能）。

有了总诀式作为心法总纲，就可以针对特定的“招式”一一破解。

破剑式 – 搞定非结构化、定制数据对象

图2: 破剑式 – 搞定非结构化、定制数据对象

要知道很多数据是不可以预先结构化的，或者是经过产品迭代过程后，预先结构化不再起作用，如图像处理等。因此非结构化的处理在物联网中显得尤为重要。

PostgreSQL是这样来应对非结构化数据场景的：首先PostgreSQL支持JSONB数据类型，该数据类型非常适合非结构化数据场景，例如传感器采集的数据以JSON格式上传；其次在定制数据对象方面，PostgreSQL开放了类型扩展和索引扩展两类接口，使用者无需关注数据库内核的实现方式，只需要关注业务本身。比如电路板的质量检测场景，使用者只需要关注焊点是否虚焊，然后再通过开放的接口将其对象化到数据库中；同时PostgreSQL中的自定义函数支持C、Python、Java等多种语言定义，扩展性极高。

破刀式 – 搞定文本、空间、时序流式数据

图3: 破刀式 – 搞定文本、空间、时序流式数据

在模糊查询、分词等文本处理方面，PostgreSQL天然支持分词的特性，包括中文分词和英文分词，性能上能够做到每秒处理千万词汇的级别，足够满足使用者的需求。

空间地理位置数据管理方面，PostgreSQL支持PostGIS和Pgrouting两种位置处理的插件，PostGIS是全球使用范围最广的地理位置信息处理插件，在美国宇航局、欧洲宇航局等企业中得到了广泛使用；Pgrouting是基于位置信息完成最短路径运算的插件。

流式处理方面，PostgreSQL 9.5以后的版本支持BRIN索引，非常适合带有时序属性的流式数据。如果按照时间来访问流式日志数据，以往需要创建B-tree索引进行范围查询或者精确匹配，但是B-tree索引会因为需要存储的较大信息量导致索引也很庞大；而BRIN记录的是每（连续）块元数据，索引变得很小。下图是两种索引之间差别详细对比：

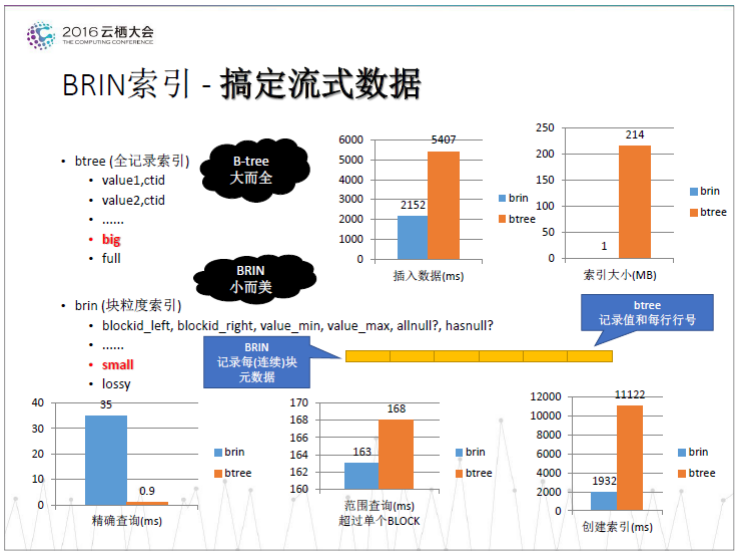

图4: BRIN索引与B-TREE 索引性能对比

破枪式 – 搞定实时流处理

图5: 破枪式 – 搞定实时流处理

实时流处理的实时性要求很高，同时传统的流式计算开发门槛高。但采用PostgreSQL，仅一条SQL就可以搞定流失实时处理。在数据源源不断地往数据库持续插入过程中，只需要定义好需要实时统计的窗口或者是流视图，数据库后台就可以实时地进行数据统计。查询流失处理结果的响应时间是在毫秒级别的。PostgreSQL在流式处理方面大大简化了开发这一环节。其处理能力相当强大，一台8G CPU的服务器每天能够处理百亿级别的流式数据。

破鞭式 – 搞定复杂查询

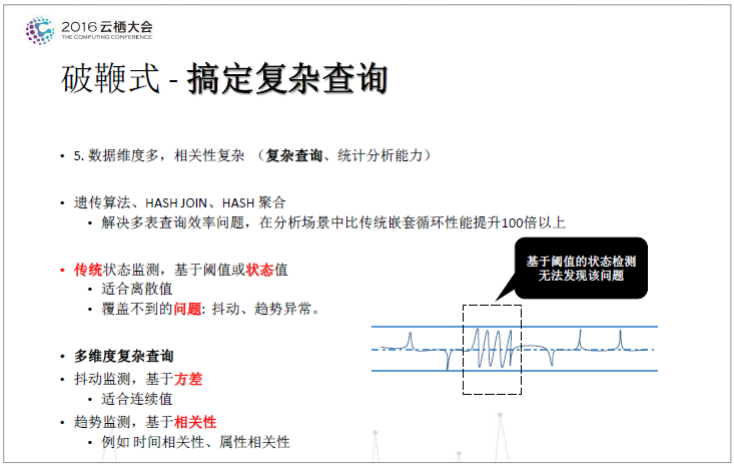

图6: 破鞭式 – 搞定复杂查询

在物联网中，因为数据维度多、相关性复杂，所以复杂查询也是一个不容忽视的问题。PostgreSQL中通过支持遗传算法、HASH JOIN、HASH 聚合，解决了多表查询的效率问题，在分析场景中比传统的嵌套循环性能提升100倍以上。

除此之外，在监测场景中，传统基于阀值或状态的监测方式是无法发现监测过程中存在抖

动、趋势异常的情况。PostgreSQL中采用基于方差的监测方式用于抖动检测；同时基于时间或属性相关性，进行趋势检测，防患于未然。

破索式 – 搞定数据分析

PostgreSQL具有强大的数据挖掘能力，可以通过一条SQL搞定数据挖掘，例如：

```
SELECT kmeans(ARRAY[x, y, z], K) OVER (), * FROM samples;
```

这条语句就可以实现聚类分析；同时PostgreSQL支持GPU，CPU并行计算，处理能力达到25GB/s，已经达到目前内存极限；此外PostgreSQL还兼容MADLib库(支持几百个机器学习库函数、对应各种数学模型)、PL/R,、PL/Python。

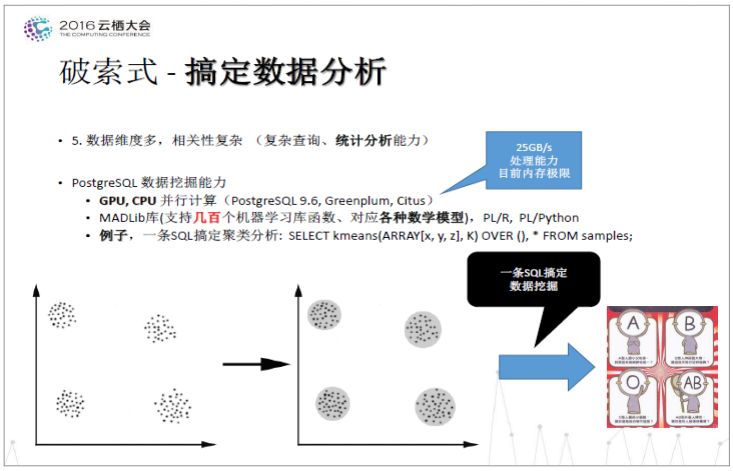

图7: 破索式 – 搞定数据分析

破掌式 – 搞定秒杀(高并发行锁竞争)

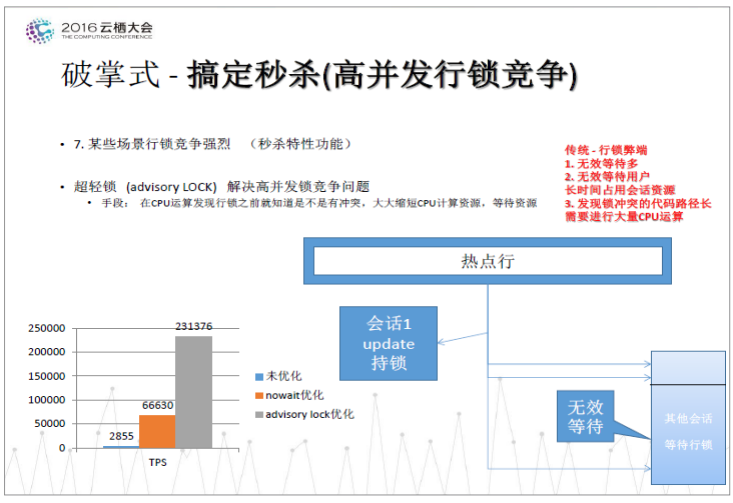

图8: 破掌式 – 搞定秒杀(高并发行锁竞争)

在物联网领域，例如秒杀等场景行锁竞争强烈。传统的行锁具有无效等待多、无效等待

用户长时间占用会话资源、发现锁冲突的代码路径长，需进行大量CPU运算等弊端。PostgreSQL提供了超轻锁（(advisory LOCK)）来解决高并发锁竞争问题，通过CPU运算发现行锁之前就知道是不是存在冲突，大大缩短CPU计算、等待资源，比如在秒杀抢手机的活动中，给定每个手机一个编号，拿到编号的用户才可以进行抢手机，这样就解决了并行度的问题，整体性能得到了近百倍的提升。

破箭式 – 搞定模糊、正则查询

图9: 破箭式 – 搞定模糊、正则查询

物联网中，对高效的模糊、相近度查询需求，较大传统查询方式是采用全表扫描的方式，百亿数据的查询响应至少是小时级别的。在PostgreSQL中，通过使用GIN R-TREE索引可以将查询时间缩短到秒级。

这里举一个模糊查询的例子，如上图所示的车牌，尽管对其中一部分做了遮挡，在PostgreSQL中，通过下几行语句，就可以轻松查出车主的个人信息：

```
select 'postregsql' % 'postgresql';
postgres=# select similarity('postregsql','postgresql');
similarity
------------
0.375
(1 row)
select * from tbl where info ~ '^???6888$';
select * from tbl where info ~ '^???688?$';
```

PostgreSQL这一特性，也是其广泛地用于公安刑侦、车牌、地址、邮箱等查询中。

破气式 – 搞定大数据处理能力

图10: 破气式 – 搞定大数据处理能力

随着数据量的增大，会衍生出非常多的问题。在PostgreSQL采取了以下几种方式处理大数据：

1.对于单机节点，采用基于CPU和GPU的计算；

2.PostgreSQL添加了FDW插件用于数据的冷热分离，可以将数据放置在Hadoop或者Spark，通过PostgreSQL提供的统一访问接口，实现HTAP（在线与离线处理一份数据）；

3.支持OLTP分库分表；

4.支持读写分离、一主多备、多副本强同步；

5.通过级联复制，解决主库压力问题和跨机房的多份数据传输问题；

6.服务端编程能力，解决move data带来的网络延迟问题；

7.支持多主复制，解决物联网地区节点和中心节点的数据相互同步问题。

接下来，针对几个特殊的特性具体分析下它们的实现过程：

FDW – 搞定HTAP

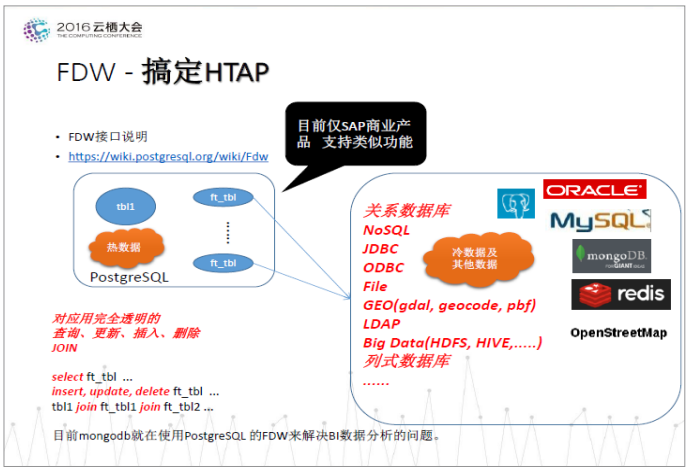

图11: FDW – 搞定HTAP

FDW目前仅在开源数据库中支持；对于商用数据库，目前仅SAP商业产品支持类似的功能。FDW可以实现数据的冷热分离和跨界访问。比如，可以将热数据存储在PostgreSQL本地，冷数据存在Hadoop或者Spark、MySQL中，通过PostgreSQL提供的统一的接口完成数据的跨界访问。目前mongodb就在使用PostgreSQL 的FDW来解决BI数据分析的问题。

数据库端编程 – 搞定网络瓶颈

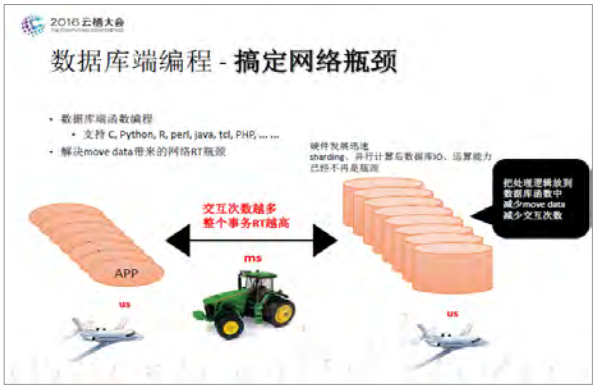

图12: 数据库端编程 – 搞定网络瓶颈

在目前的硬件条件下，普通的服务器都能达到上百核，内存达到PB级别。在这种硬件设备下，一台主机就能达到千万级别的QPS。这样就带来了一个问题，在数据库中us级别可处理的数据量，在网络中才传输可能会花费ms的时间。传统的解决方式将业务逻辑放到应用程序端实现，然后将数据库做的尽量简单。现在通过PostgreSQL，可以将代码放到数据库端，PostgreSQL提供了C、Python、R、Perl等语言的开发接口，通过数据库端编程解决数据移动带来的网络RT瓶颈。

rank化和相关性计算—— 搞定最强压缩比

图13: rank化和相关性计算 – 搞定最强压缩比

随着数据量的增大，数据的存放成本也随之增大。PostgreSQL 中提供了列存储、压缩插件，可自动整理数据压缩。

专题聚焦

PostgreSQL深入之道

策划方向：云栖社区中，PostgreSQL已经形成了自己的独特小生态。不仅是专家极为活跃，撰写了上千篇的博文，而且很多PG的用户都在向社区聚集。博文后曾有用户评价：没想到PG的问题在这里可以得到秒回。专题之外，PostgreSQL更多的功能和特性，以及实际部署中要注意的细节和要点都在社区中。

本主题链接：
https://yq.aliyun.com/topic/62

德歌：PostgreSQL独孤九式搞定物联网

找对业务G点, 体验酸爽 – PostgreSQL内核扩展指南
聊一聊双11背后的技术 – 物流, 动态路径规划
弱水三千,只取一瓢,当图像搜索遇见PostgreSQL(Haar wavelet)
用PostgreSQL支持含有更新，删除，插入的实时流式计算
PostgreSQL 内核扩展之——管理十亿级3D扫描数据(基于Lidar产生的point cloud数据)
DPostgreSQL内核扩展之 – ElasticSearch同步插件
为了部落 – 如何通过PostgreSQL基因配对，产生优良下一代
如何加快PostgreSQL结巴分词加载速度
mongoDB BI 分析利器 – PostgreSQL FDW (MongoDB Connector for BI)
关键时刻HINT出彩 – PG优化器的参数优化、执行计划固化CASE
PostgreSQL 如何潇洒的处理每天上百TB的数据增量
PostgreSQL 百亿数据 秒级响应 正则及模糊查询
PostgreSQL 百亿地理位置数据 近邻查询性能
PostgreSQL 9.6 等待事件出炉

双11前、中、后三阶段大数据计算平台全揭秘

作者：林伟

摘要： 阿里云大数据平台在双11承担了海量数据分析服务，详解今年双11的具体应用与实践。

双11备战

图1：双11的成功离不开后面大数据分析

双11的成功离不开背后大数据分析，阿里云大数据平台在双11承担了海量数据分析服务，各个部门会在计算平台上对于相关数据进行深入分析从而保障双11成功进行：通过对物流包裹预测，帮助快递公司调配仓储，使得其在双11当天能够分发6.5亿件包裹，做到兵马未动、粮草先行；对花呗授信额度进行评估，将花呗额度按照每个人风险承受额度进行相应的调整；帮助商家精准营销，对访客分群预测，设计个性化店铺首页；对消费者进行智能导购，通过分析其原始购买记录，对其进行精准化营销，提高购物体验；在双11之前，对语音模型进行了大量的训练，打造更好的语音平台，在双11当天，97%的客服电话由语音机器人来接听；此外，为了保障双11的进行，对交易安全防控、个性化推荐、商家数据服务以及营销活动反作弊都进行了训练提升。

图2：临近双11计算任务的变化

从计算平台的角度出发，可以用计算任务来量化这些准备工作：MaxCompute平台承担着阿里内部的绝大部分计算任务，日任务量为百万级别。从上图可以看出，从九月中旬开始，每日Job数目呈小幅度上升，上升幅度约为20%，这也表明了双11之前的备战是早在两个月前就已经开始了，计算平台每天都在为双11做准备。

治理数据：事半功倍

早在阿里巴巴平台业务初期，飞速增长和粗放式管理带来大量存储与计算资源的浪费，存在着大量无效的计算任务和重复的存储，例如一个新入职的小二一个上午提交了6个计算任务，运行失败，花费了18W；再比如某一个重要的应用，花了半天时间梳理自己的数据业务，把每天的花费从5W降到5千。

由于计算资源的增长落后于我们业务增长，并且我们需要降低我们成本从而能够更加高效服务于业务，因此数据治理显得尤为必要。

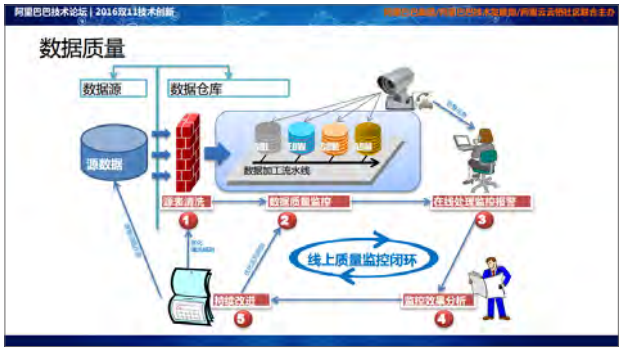

图3：数据质量

目前阿里打造了一个数据质量线上监控的闭环，将数据的访问、运行状态全部记录下来，然后通过计算平台的计算能力并行分析数据背后的关联；通过在线处理监控系统监控效果分析以及源表清洗，挖掘出数据和计算任务中的冗余数据。当发现某些数据或任务有待提高后，后续如何处理呢?

图4：治理体系

阿里目前采用的是健康分的治理体系，评价计算任务是否健康时主要从存储和计算健康度两个指标出发：存储健康度包括未管理表、废弃表、保留周期过长、同源导入等8种模型；计算健康度包括暴力扫描、产品未使用、数据倾斜、重复计算等17种模型。

两者经过一定的组合生成健康分，同时根据消耗的资源生成电子账单，然后送给业务方，使其明确所消耗的成本；此外，还设计了利益机制和操作平台来优化管理资源。

双11当天

图5：双11大屏实时效果图

上图是双11大屏的实时效果图，它能实时显示双11的实时交易额。大屏的背后支撑系统是StreamCompute（增量计算），它是阿里实时数据统计和监控的利器。双11当天，该系统使得从交易发生到媒体大屏统计出结果整个过程只耗费3秒钟；同时，它可以每秒钟处理1亿条交易记录；StreamCompute的整体性能是去年的5倍，而且整体运维过程中0故障发生。

在没有增量计算之前，整个双11的营业额统计和数据复盘处于摸索状态，双11当天发生的交易全部记录在后端，双11结束后通过批处理进行分析，最终产生报表，也就是说在报表没产生之前，是不清楚该年双11的具体状况的。这种方案无法给出实时的交易成交额，并且结果产生非常慢。

图6：业务场景——双11GMV

为了解决该问题，阿里开发了增量计算模型。通过增量计算，可以实现及时反馈，推动购

物节气氛，使得消费者互动感更好；同时能够使得双11各个参与方及时的调整策略，从而达到更好的促销效果。那么如何完成从批处理到增量计算的转化呢?

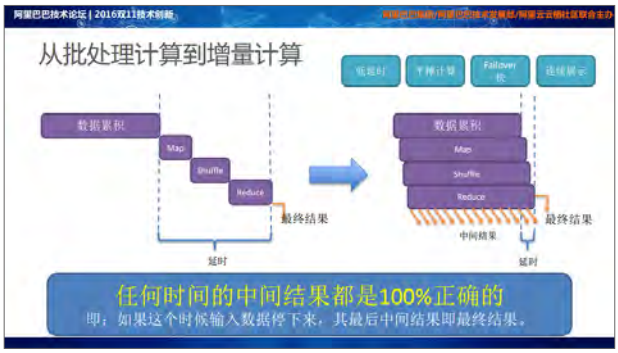

图7：从批处理计算到增量计算

如上图所示，批处理首先需要进行数据积累，数据积累完毕之后提交计算任务，经过Reduce之后产生最终结果，整个过程的时延其实是Job的Running time。

但我们希望在双11的24点就能看到最终结果，因此对其中的延时要求很高，为了满足该要求，引入增量计算。增量计算不是在数据积累完毕才开始进行任务，而是在数据积累过程中开始运算；Map、Shuffle、Reduce等阶段在数据开始时就已经调度了，整个过程中持续不断地接收数据并计算；因此最终一条记录完成时，最终结果也随之产生。这种增量计算的方式带来的好处有：

第一，低延时，最后一条记录到最终结果的产生期间的延时仅有数秒，整体的计算任务平摊到双11当天的每时每刻；

第二，Failover速度快，当出现错误时不需要从头计算，而是将中间状态进行检查，大大节省了Failover的时间；

第三，连续展示，通过增量计算可以将结果持续不断地进行展示；

第四，由于Task在持续不断地运算，因此任何时间的中间结果都是100%正确的，也就是说如果在这个时候输入数据停下来，其最后中间结果即最终结果。

图8：增量计算（统计一致性问题）

统计一致性问题是增量计算中必须解决的问题，所谓统计一致性是指任何时刻双11卖家总

数恒等于每档卖家数目。如上图所示，当A同学销售量增加到11时，红色模块加一而蓝色模块减一；当A同学销售增加15时，则绿色模块增加一，而红色模块减一。

统计一致性带来的最大好处是可以进行相对的比较，可以准确得出各个档次卖家的比例。在流式计算和增量计算中，实现统计一致性难度很高，下面来详细讲解下阿里是如何保障统计的一致性。

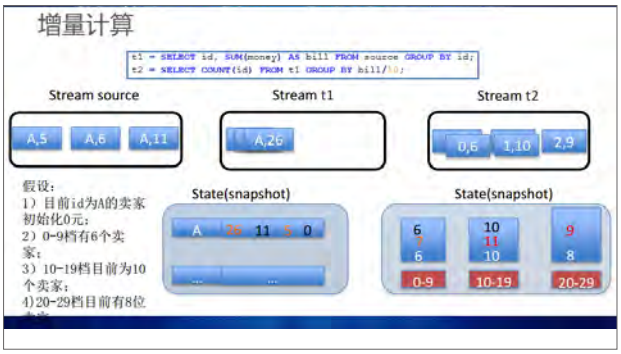

图9：阿里如何保障统计一致性

在具体双11案例中，初始化化参数为：（1）目前id为A的卖家初始化0元；（2）0-9档有6个卖家；（3）10-19档目前为10个卖家；（4）20-29档目前有8位卖家。第一阶段，A有了五元的生意，对应stream source中的（A,5），同时利用State（snapshot）记录营业额，即在stream t1中A的营业额变成5，stream t2中0-9档数目变为7。第二阶段，当A又发生6元生意时，营业额统计求和变为11，同时0-9档数目减一，10-19档数目加一；第三阶段，当A又发生15元生意时，营业额统计求和变为26，同时10-19档数目减一，20-29档数目加一。在整个交易额变化的过程中，输出都是update，无需读写操作。

增量计算的挑战中最大的挑战是任何时间的中间结果都是100%正确的，并且保证各统计数据的一致性；系统实现方面，要求所有的Operator都具有可逆性（在分布式环境需要处理跨partition的结果调整）。为了保障可逆性，需要state来记住原来产生的中间结果，能够快速定位到需要调整的value，进行正向和逆向操作；此外，增量还具有其他流计算的普遍问题，如流控、数据倾斜、容错和延时等等。

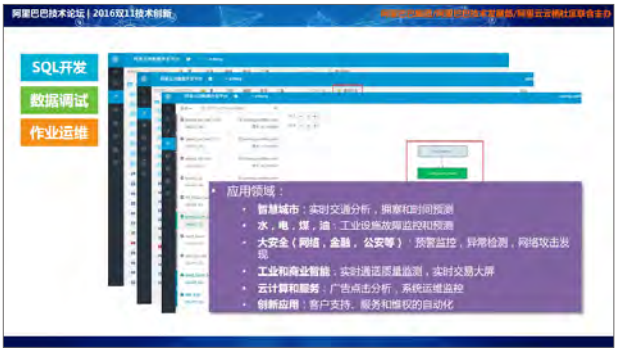

图10：增量计算提供的能力

在增量计算中还提供了很好的SQL开发界面，便于业务方开发；同时还提供了完善的数据调试手段以及丰富的作业运维，目前的应用场景主要包括：

智慧城市：实时交通分析，拥塞和时间预测；

水、电、煤、油：工业设施故障监控和预测；

大安全（网络，金融， 公安等）：预警监控，异常检测，网络攻击发现；

工业和商业智能：实时通话质量监测，实时交易大屏；

云计算和服务：广告点击分析，系统运维监控；

创新应用：客户支持、服务和维权的自动化。

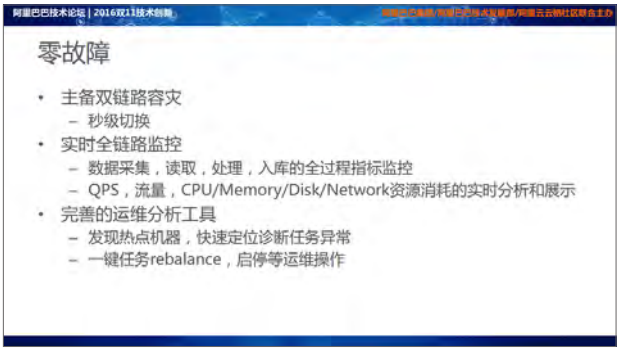

图11：保证零故障

为了保障双11期间的零故障，阿里也提前做了很多准备：首先采用了主备双链路容灾，实现秒级切换；同时进行全链路监控，对数据采集、读取、处理、入库的全过程指标监控，对QPS、流量、CPU/Memory/Disk/Network资源消耗的实时分析和展示，充分探究潜在问题；此外，还为双11配备了完善的运维分析工具，能够分析发现热点机器，快速定位诊断任务异常，以及进行一键任务rebalance、启停等运维操作。

双11后海量数据分析

双11产生了海量的数据，那么如何在双11之后进行复盘对数据进行分析，确保各类对账单能够按时按质的输出呢？这就依赖于后端的计算平台——MaxCompute。

图12：后端计算平台-MaxCompute

MaxCompute承载了阿里巴巴集团所有的离线计算任务，是集团内部核心大数据平台。截

止到目前支撑着每日百万级规模的作业，整个系统拥有数万台机器，单集群规模上万，存储已经到达了EB级别，每天有数千位活跃的工程师在平台上做数据处理。

MaxCompute目前具有两大成就：第一是在今年双11创纪录处理了180PB的数据；第二是100TB数据排序耗时仅为377s。

图13：MaxCompute：HBO

在MaxCompute中，大量的任务具有周期性，每天相似的查询会给优化器带来巨大机会，因此可以基于历史进行特定的优化，对每天提交的查询进行聚类，把以前运行数据作为Hint来帮助未来的相似的查询。新的查询首先经过相似判断，如果是相似查询，则进行Hint注入，帮助进行该次优化，当数据变化不大时，相当于查询预热。

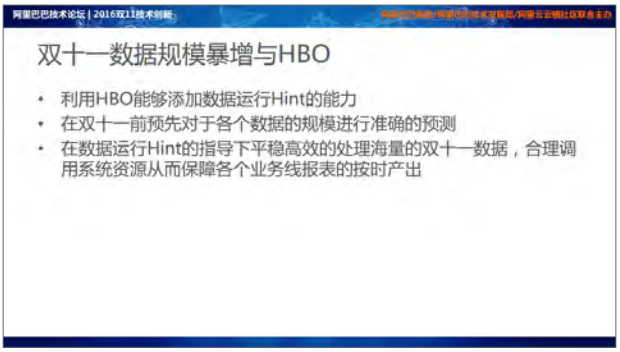

图14：双11数据规模暴增与HBO

在双11当天数据的暴增进而导致HBO的效果下降。为了解决这个问题，在双11到来前利用多种模型预先对各个数据的规模进行准确的预测，同时利用HBO能够添加数据运行Hint能力帮助双11当天的任务按照合理的配置调用资源，进而保障各个业务线报表的按时产出。

MaxCompute：全局调度

MaxCompute已经达到了物理集群的上限，单集群已达到上万级别，因此数据处理平台一定是跨地域搭建的，地域与地域之间采用昂贵的总线和专用网络连接。在双11海量数据产生时如何充分有效地利用跨地域带宽，做到带宽和延时的平衡呢?

图15：MaxCompute全局调度

我们设计了一个全局调度方案，使得用户无需关心数据分布的具体位置，由系统帮助其访问。如上图案例所示，当A提交数据Update操作后，距离其最近的集群中存放着最新的数据；当B访问该数据时，系统会自动识别距离其较远的集群内的数据是最新数据，然后利用有限的带宽进行数据复制，维持数据的一致性。这个过程中有很多种选择，可以采用远程读、Replicate等多种模式；同时还需要充分考虑带宽，任务完成时效需求；需要进行全局分析，动态预先调整。

专题聚焦

数加1周年分布式技术全解

简介：阿里云大数据平台(数加)一周年专题策划，特别对分布式技术如存储、调度、计算到机器学习等方面进行了分析，还邀请了上海云贝、杭州以数科技、杭州玳数科技、袋鼠云、朗新科技等CTO，讲述大数据应用场景的故事。

本主题链接：
https://yq.aliyun.com/topic/79

林伟：大数据计算平台的研究与实践
高可用的大数据计算平台如何持续发布和演进
盘古：阿里云飞天分布式存储系统设计深度解析
10年老兵带你看尽MaxCompute大数据运算挑战与实践
需求、系统、动机如何满足？阿里云推荐系统架构深度解析
流计算StreamCompute
钱正平：大规模实时计算及其在阿里的应用与创新
机器学习为您解密雾霾形成原因
人口普查统计案例
机器学习应用没那么难，这次教你玩心脏病预测
实践！如何用阿里云的机器学习得出泰坦尼克号沉船事件中谁有更大的概率获救
机器学习算法的离线调度实现-广告CTR预测
文本分析算法实现新闻自动分类
听说啤酒和尿布很配？本期教你用协同过滤做推荐

2016阿里安全峰会：“安全攻防”烧脑博弈全解读

摘要：在2016阿里安全峰会的第二天的“安全攻防”专题论坛，漏洞盒子高级安全研究员陆柏廷、安恒信息高级安全研究员郑国祥等专家为我们带来了精彩演讲。

在2016阿里安全峰会的第二天的“安全攻防”专题论坛，漏洞盒子高级安全研究员陆柏廷、安恒信息高级安全研究员郑国祥、四维创智技术副总裁司红星、UnicornTeam无线通信安全研究员张婉桥、猎户实验室首席安全研究员潘立亚、四叶草首席安全官朱利军、东巽科技CTO李薛、腾御安信息基础架构安全运维杨旭为我们带来了精彩演讲。

陆柏廷：典型业务逻辑漏洞挖掘

陆柏廷，漏洞盒子高级安全研究员，专注业务安全、安全开发和移动安全，负责大量安全测试项目，有丰富的安全服务及咨询经验，并参与网藤风险感知系统的研发，对安全测试和安全开发有较深的积累。

陆柏廷由用户认证这个典型业务场景的漏洞挖掘技巧，讲述了安全人员与开发人员之间的烧脑博弈，主要包括用户认证处可能遇到的各种逻辑问题及产生原因，最后总结业务逻辑漏洞高效的挖掘方法。

PDF下载：典型业务逻辑漏洞挖掘

郑国祥：威胁情报在网络犯罪侦查中的落地应用

郑国祥，杭州安恒信息技术有限公司高级安全研究员，从事web应用安全研究，擅长漏洞挖掘，源代码分析。长期研究源码的静态和动态分析方法，熟悉各种安全web框架，防御绕过技术等。为多个src平台提供大量安全漏洞，2014年获tsrc最佳专业奖，2015 腾讯tsrc漏洞之王，发现多个CVE漏洞严重漏洞，其中包括了s2-029, s2-032等漏洞。

郑国祥介绍了web框架安全漏洞的fuzzing测试，利用静态代码流程分析，结合fuzzing跟动态检测机制挖掘框架0day，突破漏洞利用限制将鸡肋漏洞转变成严重漏洞。

PDF下载：WEB框架0day漏洞的发掘及分析经验分享

司红星：新一代自动化渗透平台的设计与实现

司红星，四维创智技术副总裁，常年从事APT攻击与防御技术研究，擅长网络攻防技术，具备丰富的实战经验，思路猥琐，手法飘逸，在公司带队研发多个信息安全产品，将自动化渗透思路带入企业安全防御体系，致力于打造一个具备高级攻击能力的自动化渗透平台，让每一家公司都能雇得起高质量“白帽子”。

司红星谈到，新时代下，互联网安全越来越受重视，然而人才缺乏，信息安全工程师缺乏高效的渗透测试装备，传统企业雇佣不起高质量白帽子。本议题中，司红星从自动化渗透出发，讨论了如何建立一套能达到中上等渗透测试水平的自动化渗透平台，提升安全从业者工作效率，降低企业安全运维成本。

PDF下载：新一代自动化渗透平 台的设计与实现

张婉桥：短距离无线系统与航空无线电系统攻击

张婉桥，UnicornTeam无线通信安全研究员。张婉桥以《短距离无线系统与航空无线电系统攻击》为题进行了分享。

PDF下载：短距离无线系统与航空无线电系统攻击

潘立亚：漏洞与数据的奇点临近

潘立亚，猎户实验室首席安全研究员，擅长web攻防、建模分析。他介绍了在漏洞挖掘中分析保护数据（积极防御）的规律方法并且根据积累的数据对攻击和防御研究创作。

PDF下载：漏洞与数据的奇点临近

朱利军：浅谈“互联网＋”漏洞研究之道

朱利军，四叶草首席安全官，国内顶级安全信息专家，接触和从事安全行业至今8年。现担任CSO（首席安全官）职务兼CloverSec Labs实验室负责人。对复杂环境下的网络攻防有深入研究，负责与组织与公司相关的网络攻防大赛，各次大赛均得到业界各人士的一致好评，并创造了国内单场比赛参赛人数最多的业绩。主持并带领团队进行了国内知名厂商嵌入式设备漏洞挖掘，以及对全球25款安全杀毒软件进行漏洞研究，发现高危漏洞若干。

朱利军从“互联网＋”的安全威胁入手，介绍“互联网＋”下都存在哪些安全威胁和“互

联网＋”环境下各类安全威胁的漏洞研究思路。针对Web、安全防御软件、工业控制系统、智能联网设备等漏洞研究的方法与思路。然后站在黑客的角度思考，如何快速定位与发现“互联网＋”中的漏洞，最后以几个实际的漏洞挖掘实例介绍漏洞研究的灵活思路。

李薛：机器学习在恶意样本检测方面的实践之路

李薛，现任东巽科技CTO，中国科学基金委员会评议人。李薛曾主持过多项部委级安全科研项目，连续两年担任XDef峰会演讲嘉宾，是中国较早的一批研究网络攻防技术团队的成员，在网络安全研究方向拥有15年的经验积累，同时具有8年技术团队管理经验，现就职于东巽科技公司，带领团队从事APT研究工作。

他描述了机器学习在恶意样本检测方面，如何一步一步从零走过来，如何在问题面前做选择，如何逐步改进，如何跨坑，如何落地，还有需要重点考虑和注意的要点，以及在迭代改进的研究工作流程。

PDF下载：机器学习在恶意样本检测方面的实践之路

杨旭：Debian GNU/Linux安全合规之路

杨旭，腾御安信息基础架构安全运维。杨旭以《Debian GNU/Linux安全合规之路》为题进行了分享。

PDF下载：Debian GNU/Linux安全合规之路

专题聚焦

聚力 · 赋能——2016阿里安全峰会

策划方向：2016阿里安全峰会的主题是聚力 · 赋能。当前网络安全迎来了新的发展机遇，但安全挑战也日益严峻。人的第一需求是生存，企业的第一需求是发展，安全是为了更好地保障生存与发展。因此，安全应该是对客户的赋能：让客户具备更好的威胁应对与风险控制能力，以顺利实现生存和发展的目标。与此同时，攻守本身的不平衡、暗黑力量的不断壮大，使得我们迫切需要携起手来、相互赋能，实现高效的联防。这里整理了主会与分论坛的所有资料。

本主题链接：
https://yq.aliyun.com/topic/47

2016阿里安全峰会：“安全攻防”烧脑博弈全解读
2016阿里安全峰会：“电商金融系统与业务安全”全解读
阿里CEO张勇：安全是中国互联网生态的共同基石
蚂蚁金服副总裁陆杰讯：搭建安全统一战线，不仅要防守，还要进攻
阿里集团首席风险官刘振飞：阿里安全九字方针“轻管控、重检测、快响应”
阿里集团商家事业部总经理张阔：商业生态需要安全赋能
Fireeye前副总裁卜峥：不知攻焉知防，打造“3C的安全体系结构”
肖新光：熊猫的伤痕——几例中国遭遇APT攻击的案例分析
杜跃进：数据安全不仅是数据不被偷走，而是没有滥用
肖力：阿里云的安全策略是责任共担，与用户“并肩作战”
蚂蚁金服安全产品技术资深总监冯春培：用生态的力量解决安全生态的问题
安恒信息安全专家刘志乐：复杂网络环境下的大数据安全态势感知
阿里巴巴安全技术运营资深专家张玉东：从生态角度打造新的安全

体系结构

2016阿里安全峰会：风声与暗算，无中又生有：威胁情报应用的那些事儿

2016阿里安全峰会：云中行走——漫谈云端安全

2016阿里安全峰会：封闭的冲突与开放的和平：白帽子与SRC

2016阿里安全峰会：秘在其中：做好数据与信息的安全管理

2016阿里安全峰会：解读安全人才缺乏困境破解之法

2016阿里安全峰会：变革环境下的IT治理与网路安全

2016阿里安全峰会：电子取证：静静聆听那些真相

2016阿里安全峰会：关于移动安全的那些事儿

罗龙九：云数据库十大经典案例分析

摘要：在阿里巴巴在线峰会上的第二天，来自阿里云资深DBA专家罗龙九给大家带来了题为《云数据库十大经典案例分析》的分享。罗龙九以MySQL数据库为例，分析了自RDS成立至今，用户在使用RDS过程中最常见的问题，包括：索引、SQL优化、锁、延迟、参数优化、连接数、CPU、lops、磁盘、内存等。

罗龙九以MySQL数据库为例，分析了自RDS成立至今，用户在使用RDS过程中最常见的问题，包括：索引、SQL优化、锁、延迟、参数优化、连接数、CPU、lops、磁盘、内存等。罗龙九通过对十大经典案例的总结，还原问题原貌，给出分析问题的思路，旨在帮助用户在使用RDS的路上少一些弯路，多一些从容。以下为整理内容：

案例一：索引

图1

今天之所以将索引放在第一位进行分享，是因为索引的问题出现频率是最高的。常见的索引问题包括：无索引、隐式转换两类。其中隐式转换是由SQL传入的值和表结构定义的数据类型不一致引起；或者是表字段定义的collation不一致导致多表join的时候出现隐式转换。无索引的情况会导致全表扫描；隐式转换会导致索引无法正常使用。

图2

在使用索引时，我们可以通过explain（extended）查看SQL的执行计划，判断是否使用了索引以及发生了隐式转换。由于常见的隐式转换是由字段数据类型以及collation定义不当导致，因此我们在设计开发阶段，要避免数据库字段定义，避免出现隐式转换。此外，由于MySQL不支持函数索引，在开发时要避免在查询条件加入函数，例如date(gmt_create)。最后，所有上线的SQL都要经过严格的审核，创建合适的索引。

案例二：SQL优化

图3

SQL优化是很多使用者都需要面对的问题。我们在不断地优化、调试过程中总结了三类SQL优化的最佳实践，分别是分页优化、子查询优化、查询需要的字段。

分页优化

这条语句是普通的Limit M、N的翻页写法，在越往后翻页的过程中速度越慢，这是由于MySQL会读取表M+N条数据，M越大，性能越差。我们通过采用高效的Limit写法，可以将上述语句改写成：

```
select t1.* from buyer t1,
(select id from buyer sellerid=100 limit 100000，5000) t2
where t1.id=t2.id;
```

从而避免分页查询给数据库带来性能影响。需要注意一点是，这里需要在t表的sellerid字段上创建索引，id为表的主键。

子查询优化

子查询在MySQL5.1、5.5版本中都存在较大的风险。这是一段典型子查询SQL代码：

```
SELECT first_name
```

```
FROM employees
WHERE emp_no IN
(SELECT emp_no FROM salaries_2000 WHERE salary = 5000);
```

由于MySQL的处理逻辑是遍历employees表中的每一条记录，代入到子查询中去 。所以当外层employees表越大时，循环次数也随之增多，从而导致数据库性能的下降。这是我们改写子查询之后的SQL代码：

```
SELECT first_name
FROM employees emp,
(SELECT emp_no FROM salaries_2000 WHERE salary = 5000) sal
WHERE emp.emp_no = sal.emp_no;
```

首先将子查询的结果放到临时表内，再去和employees表做关联。此外，使用者也可以选择使用Mysql 5.6的版本，避免麻烦的子查询改写。

查询需要的字段

在访问数据库时，应该尽量避免使用SELECT *查询所有字段数据，只查询需要的字段数据。

案例三：锁

图4

在使用数据库时，每个人或多或少都会碰到锁的问题。在设计开发阶段，我们需要注意这三点问题：一是避免使用myisam存储引擎，改用innodb引擎；二是注意避免大事务，这是因为长事务导致事务在数据库中的运行时间加长，造成锁等待；三是选择将数据库升级到支持online ddl的MySQL 5.6版本。

在管理运维阶段，我们可以从四点出发搞定锁的问题：

1.在业务低峰期执行上述操作，比如创建索引，添加字段；
2.在结构变更前，观察数据库中是否存在长SQL，大事务；
3.结构变更期间，监控数据库的线程状态是否存在lock wait；
4.RDS支持在DDL变更中加入 wait timeout。

案例四：延迟

图5

由于数据库架构大多是主备的方式，延迟便成了一个常见的问题。产生延迟的原因有很多，例如在只读实例架构中，主备节点间MySQL原生复制实现数据同步方式会天然导致延迟的产生。此外，create index、repair等常见DDL操作、大事务、MDL锁以及资源问题都会导致延迟的出现。

图6

处理延迟问题，需要具有清晰的排除思路：一看资源是否达到瓶颈；二看线程状态是否有锁；三判断是否存在大事务。同时我们还可以通过使用innodb存储引擎、将大事务拆分为小事务、DDL变更期间观察是否有大查询等具体最佳实践降低延迟。

案例五：参数优化

图7

我们曾经遇到这样一个案例，某金融客户在将本地的业务系统迁移上云后，在最高配置的RDS上运行时间明显要比线下自建数据库运行时间慢1倍，进而导致客户系统出现割接延期的风险。对于这类案例的分析，根据以往的经验，可以从以下三点出发：

1. 首先查看数据库是否是跨平台迁移（PG->MySQL、ORALCE->MySQL）；
2. 其次查看是否是跨版本升级(MySQL:5.1->5.5、5.5->5.6)，不同版本之间是有差异的；
3. 如果上述两点都不存在，则需查看具体的执行计划、优化器、参数配置、硬件配置。

图8

如果SQL从云下迁移到云上或者从一个版本迁移到另一个版本的过程中出现性能问题时，要保持清晰的排查思路：从SQL执行计划到数据库版本和优化器规则，再到参数（包括Query_cache_size、Temp_table_size）配置和硬件配置等一一进行排查。曾经看到这样一个案例，一个用户使用默认的mysql配置跑线上应用，db所在的主机的内存有500G，但是分配给MySQL的内存确是默认的128M，导致了整个系统的性能下降。

案例六：CPU 100%最佳实践

图9

导致CPU 100%的三大因素分别是：慢SQL、锁和资源。对于慢SQL问题：我们可以通过优化索引或者通过避免子查询、隐式转换以及进行分页改写等措施从根上解决该问题。对于锁等待问题：可以通过设计开发和管理运维优化锁等待。对于资源问题：可以通过参数优化、弹性升级、读写分离、数据库拆分等方式优化。

案例七：Conm 100%

图10

导致Conm 100%的三大因素分别是慢SQL、锁、配置。对于慢SQL问题：解决方案类似于处理CPU 100%，同样是通过优化索引或者通过避免子查询、隐式转换以及进行分页改写等措施从根上解决该问题。对于锁等待问题：同样可以通过设计开发和管理运维优化锁等待。对于配置问题：我们需要合理规划数据库上的连接数的使用，避免客户端连接池参数配置超过实例最大连接数的情况出现。此外，还可以通过弹性升级RDS的规格配置来满足客户端需要的连接数。

案例八：Iops 100%

图11

Iops 100%也是一个很常见的问题。导致Iops 100%的原因也可以分为慢SQL问题、DDL、配置问题三类。对于慢SQL问题：解决方案同样类似于处理CPU 100%问题，通过优化索引或者通过避免子查询、隐式转换以及进行分页改写等措施从根上解决该问题。对于DDL问题：一定要避免并发进行create index、optimze table、alter table add column等操作；同时这些操作最好在业务低峰期进行。对于配置问题：可以通过弹性升级RDS的规格配置解决。

案例九：disk 100%

图12

磁盘空间由数据文件、日志文件和临时文件组成。对于数据空间问题：由于数据文件的索引和数据是放在一起的，当对表删除数据后可以采用optimize table收缩表空间，同时删除不必要的索引；对于写多读少的应用，可以使用tokudb压缩引擎进行表压缩。对于日志空间问题：首先我们需要减少大字段的使用；其次可以使用truncate替代delete from。对于临时空间问题：一是可以适当地调大sort_buffer_size；二是可以创建合适索引避免排序。

案例十：mem 100%

图13

当内存使用率达到100%时，操作系统会kill掉MySQL进程，从而导致业务的中断。因此，我们需要明确地了解数据库的内存使用详情。数据库内存主要由Buffer pool size 、Dictionary memory、Thread cost memory三部分组成。对于Buffer pool size问题：首先，我们可以通过创建合适的索引，避免大量的数据扫描；其次，我们需要去除不必要的索引，降低内存的消耗。对于Thread cost memory问题：一方面，我们可以通过创建合适的索引避免排序；另一方面，在查询数据时，我们只查询应用所需的数据，避免所有数据的查询。对于Dictionary memory问题：当表被访问打开后其元数据信息是存储在Dictionary memory之中的，过度的分表会导致内存的大量占用，因此分表时要注意把握分寸，不多过度分表，曾经看到一个数据库中创建了十几万张表。

关于分享嘉宾：

罗龙九，阿里云资深DBA专家，有着丰厚的DBA经验，经历阿里历年双11考验，负责阿里云RDS线上稳定以及专家服务团队，积累了6年对阿里云数据库用户的运维、调优、诊断等丰富的经验。

专题聚焦

阿里巴巴在线技术峰会

策划方向：首届阿里巴巴在线技术峰会（Alibaba Online Technology Summit）于7月19日-21日 20:00-21:30 在线举办。峰会邀请到了阿里集团9位技术大V，分享电商架构、安全、数据处理、数据库、多应用部署、互动技术、Docker持续交付与微服务等一线实战经验，解读最新技术在阿里集团的应用实践。

本主题链接：
https://yq.aliyun.com/topic/52

蒋晓伟：Blink计算引擎
罗龙九：云数据库十大经典案例分析
李金波：企业大数据平台仓库架构建设思路
易立：从Docker到容器服务 ——Docker 云端实践之路
郑恩阳：电商互动营销的技术实现
魏鹏：基于Java容器的多应用部署技术实践
郭东白：基于大数据的全球电商系统性能优化
方超：阿里聚安全在互联网业务中的创新实践
何登成：AliSQL性能优化与功能突破的演进之路

快捷支付时代，支付宝如何保障亿级用户的性能稳定?

作者：钟鹞

摘要：从产品和架构演进、性能及稳定性挑战与优化实践、超级APP运维体系、架构上的容灾规划四个方面分析支付宝如何保障亿级用户的性能稳定。

本文根据蚂蚁金服钟鹞在《蚂蚁金服&阿里云在线金融技术峰会》上的分享整理而成。从产品和架构演进、性能及稳定性挑战与优化实践、超级APP运维体系、架构上的容灾规划四个方面分享了支付宝APP亿级用户的性能稳定性优化及运维实践。

支付宝介绍

刚开始的支付宝内容非常单薄，只有转账、缴费、话费充值。经过五六年的努力，现在的支付宝已经变成了整个蚂蚁金服承载金融业务的一个重大平台，承载着非常丰富的场景，也将作为未来生活互动的大平台。在未来，支付宝希望能够输出金融的能力，和大家一起完成普惠金融的梦想。

图1

APP架构演进

在产品发生巨大变化的同时，技术架构上也发生了巨大的变化。总体来看分为三个阶段：

第一个阶段集中在2013年以前，支付宝从架构上来看是一个简单的分层、单体应用，上面是很多业务模块，下面是工具库；

图2

2013年到2015年，最大的变化就是支付宝变成了一个服务化、模块化的应用，这样支付宝可以被多个团队并行开发；

2015年之后，我们发现支付宝不仅需要面对支付宝内部的各个部门做功能模块的开发，而且各个行业的应用都会往支付宝上去投放，整体来看是一个多应用的生态。技术上主要实现了开放和动态化，动态化是指不同的应用能够很快速的开发并且投放到支付宝上来，而且能够分发到不同需要的用户上去。对于超级APP来说，关键的一个设计是高可用、高性能、高灵敏度。

支付宝整体的架构和其他公司相比有很大的区别，支付宝采用混合型架构，包括：支付型架构、移动互联网金融型架构、生活互动型架构。

技术挑战

业务复杂性

支付宝在用户规模、功能的复杂度上远远超过了其他的APP。

图3

设备多样性

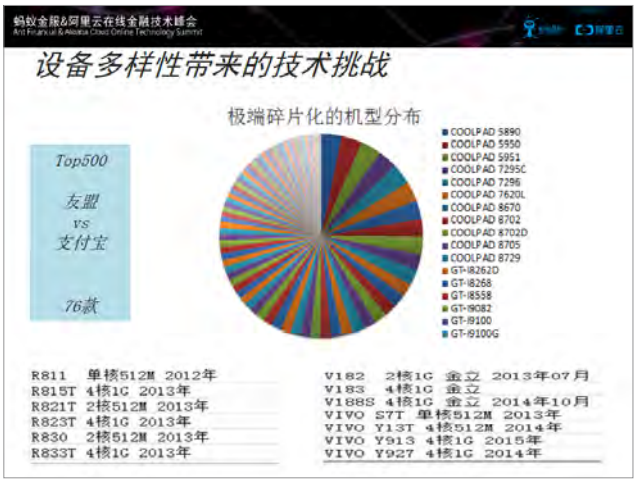

图4

性能问题的范畴

在支付宝内部有一个专门的团队负责支付宝性能优化，狭义的性能是指启动时间、流畅度、卡顿。除了传统的性能问题之外，还有流量、电量、内存、存储等广义的性能问题。随着业务的发展，未来广义性能问题会变得越来越重要。

图5

卓有成效的性能优化实践

性能优化：一个应用达到很大的规模之后，以前讨论的单点优化的意义变得不大了，因为能优化的都已经优化的差不多了。支付宝内部有一个模块化的容器quinox可以支撑大规模同时的并行开发，其最大的优化就是改成了按需加载。通过线程治理，重新梳理了线程池设计、资源管控、CPU调度的思想。性能优化的一个重要部分是虚拟机dalvik的调优，包括关闭jit让应用的第一次启动加快、去dexopt。主线程优先级调整让主线程更快的跑起来，

对其他线程的nice值进行调整，启动流程重构，引入pipeline机制把启动过程按序区分出来，使启动过程变得干净的同时方便监各部分控耗时情况。

图6

电量优化：系统兼容性优化、业务优化，技术基础优化，特别是Wakelock，一个应用如果申请了Wakelock但是由于某些原因没有释放，这时候就会发现CPU一直没有休眠导致手机的耗电量一直在增加。

流量优化：所有的资源都做了增量/差量下载，让以前需要下载的全量的包变得小很多，对支付宝的应用进行按需下载，底层技术上做了RPC及整个底层网络协议的优化。

内存优化：主要把一些很耗内存的图片转移到Native上，虽然总的内存不会变，但是流畅度有很大的改观。并且对内存泄漏、对象占用做了深入的梳理。

存储优化：对so编译使用了STL共享库，Bundle中非必需的lib转移到assets上。研究新的压缩算法对日志进行压缩。

电量优化

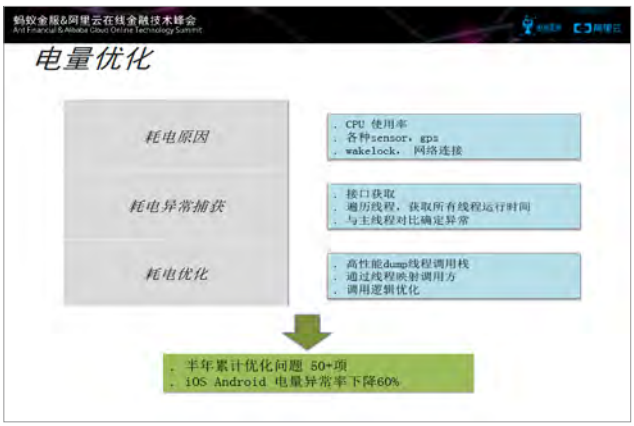

图7

怎么去衡量电量是一个专业的问题。支付宝采用两个指标衡量耗电：支付宝现在的耗电排名是多少？支付宝客户端耗电的占比是多少？经过排查，发现耗电的原因包括：CPU的使用率，过载线程会导致CPU的占用时间变长；各种sensor，gps；wakelock，网络连接。先进行耗电异常捕获然后再进行耗电优化。比如，当CPU耗电过多时我们希望知道哪个线程引起的，当我们发现支付宝的耗电量排名第一并且耗电占比10%甚至20%以上时，我们就把支付宝客户端里面所有线程dump出来，当dump线程调用栈的时候，操作系统会告诉你每一个线程耗了多少CPU的时间，这是一个非常重要的线索，我们可以从中定位出是哪个线程的哪行代码在消耗你的CPU。

流量优化

图8

具体原理和电量优化类似，引入了一个流量指数的概念来评价用户在使用支付宝客户端时的流量消耗。参数包括总流量的和、不同访问请求重复的次数。

内存优化

图9

比较大的一个技术改进是把整个支付宝客户端的内存在底层归属到不同的模块中，然后分析每个模块的占用是否合理。

稳定性

稳定性问题包括：启动闪退、启动卡死、Crash、ANR。

Crash优化

图10

随着用户量的增长，传统的鉴定Crash的方法需要做一些调整。我们需要关注的问题是每天有多少人因为稳定性问题使用不了支付宝。所以我们把以前Crash的单纯定义做了一些细化，将总体的Crash细化为一次性的Crash和持久性的Crash，期望将持久性的Crash降到很低。另外，考虑到后台Crash对用户的影响还是可控的，所以我们还将Crash分为前台Crash、后台Crash。除了Java层的闪退之外，对Native层的闪退关注是更多的。

稳定性优化

图11

思路与其他一样：监控、诊断、修复。为了把对用户的影响降低到最小，对启动过程中的闪退做了一些容灾措施，比如，发现一个用户连续三次都启动不了，则自动清理非隐私数据保证下次启动顺畅。

超级APP的运维体系

图12

线上异常监控

图13

在客户端，把一些关键的流程上通过切片技术布置监控点。在服务器端，当问题出现时能够将其模块化并且自动报警，通知不同的团队其健康状况。在数据分析上，利用各种图表、维度将数据展现出来，让用户能够看到均值、分布、尾部，方便出现问题时进行回滚。

电量指数的计算

图14

Android4.4之后，系统将电量权限收掉之后，电量计算成为了一个难题。目前的方法是模仿整个Android系统计算电量的公式重新做了一遍，即首先从BatteryStats.bin中拿到每一个维度的权重，将维度和权重结合就可以算到Android系统的耗电量。

快速定位与诊断

图15

每个部分出现问题的时候，比如CPU出现问题的时候把线程的调用栈打出来，并且获得线程消耗的时间，这样才能得知那个线程消耗时间比较多。

多层次的动态化技术

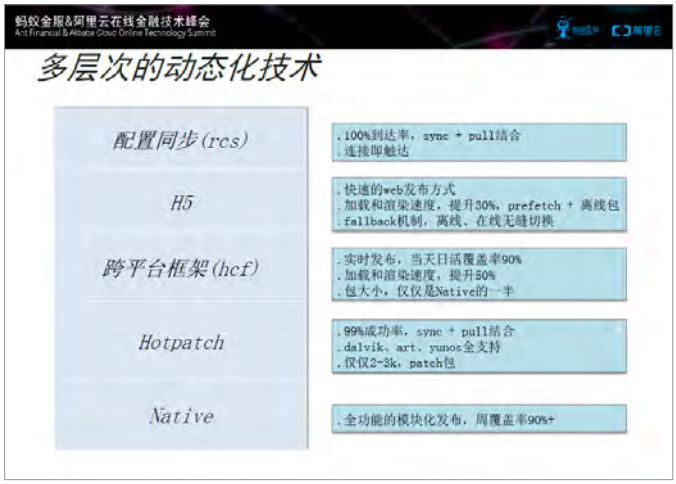

图16

支付宝的动态化技术不是单点的，而是层次化的。从上到下分为五个层次：

- 配置同步（rcs）：很多场景只需要一个配置就行，这是最基础的一个动态化，但是很有学问，怎么保证配置能够快速到达客户端？中间怎么去同步？
- H5：用来做开放和承接外面的业务，以及支付宝内部要求不是很高但是动态化要求比较高的业务。
- 跨平台框架（hcf）：如果对性能有要求的话，使用跨平台框架，用来兼顾开放、动态化、性能的要求。
- Hotpatch：用来支持代码修复。
- Native：支持把整个Bundle替换掉。

架构的容灾能力

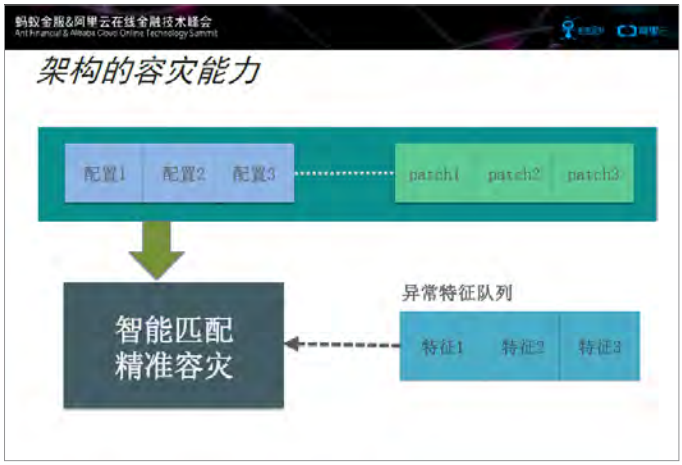

图17

在服务器端出现问题时，可以通过回滚等方法及时修复，但是在客户端出现问题时会比较麻烦，所以需要从架构方面对整个容灾能力做新的规划。把客户端出现的各种异常做一个抽象，然后提取出来各种特征，在服务器端针对这些特征配置相应的应对方法。

专题聚焦

蚂蚁金服&阿里云在线金融技术峰会

专题简介：蚂蚁金服&阿里云在线金融技术峰会于8月30日-31日在线举办。8位蚂蚁金服和阿里云技术大V，从蚂蚁无线技术、大规模机器学习、金融领域大数据创新、云数据库OceanBase、蚂蚁开放平台等维度，分享了蚂蚁从FinTech到FinLife的技术探索思考，展现技术在互联网及金融领域的创新实践应用。

本主题链接：
https://yq.aliyun.com/topic/59

云计算时代
负载均衡如何优化才能让性能起飞?

作者：金帅

摘要：在云计算时代，我们输出计算能力会像水和电一样方便。作为网络流量的入口，负载均衡技术在云计算中的地位至关重要，如何对负载均衡进行高性能、高可用的优化?

本文根据阿里云飞天八部金帅在大流量高并发互联网应用实践在线峰会上的题为《双11技术揭秘——负载均衡性能优化演进之路》的演讲整理而成。以下是精彩内容整理：

在云计算时代，我们输出计算能力会像水和电一样方便。提到云计算时，大家可能更多会想到计算相关的云产品，比如云主机ECS、关系型数据库RDS、大数据处理平台ODPS，但其实负载均衡在云计算里面的地位是至关重要的，因为它是网络流量的入口。互联网时代，计算资源、服务器、手机、电脑、物联网设备需要网络去连在一起。云计算时代，分布式计算往往意味着一个集群里面有很多计算节点，怎么保证服务的请求会均匀分布在计算节点上面同时对外提供服务呢? 这是负载均衡需要解决的一个课题。

云上的“双11”

2015年的“双11”第一次有阿里云参与进来，是云上的“双11”。“双11”的阿里云主要有以下几个部分：阿里云三大件，即云服务器、负载均衡、RDS云数据库，这三大件是云计算的基础组件。

图1

SLB负载均衡的“双11”提出了很多问题：如何实现快速部署? 如何提供足够的性能? 如何提供高可用的服务? 如何提供足够的容量?

负载均衡简介

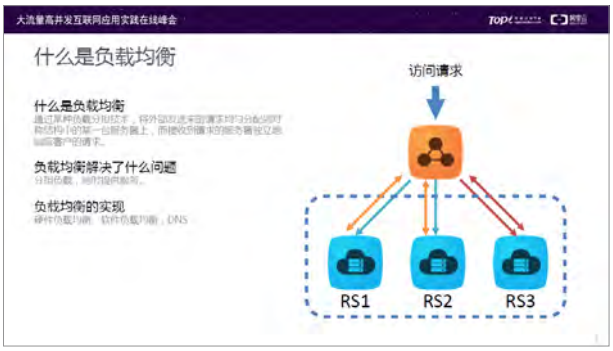

图2

负载均衡是指通过某种负载分担技术，将外部发送来的请求均匀分配到对称结构中的某一台服务器上，而接收到请求的服务器独立地回应客户的请求。通俗的讲，就是当访问请求变多时，需要更多的服务器来响应请求、对外提供服务，这时候需要有一个负载均衡器通过一定规则把请求分发在多台服务器上，横向拓展了服务的性能。负载均衡可以通过硬件或者软件来实现，传统的负载均衡还可以通过DNS实现，但是DNS的时效性不好。

硬件负载均衡器是一个重要的角色。十年以前，负载均衡大部分是由负载均衡器来实现的，最出名的厂商是F5。负载均衡器的优点是：有专门的团队来提供开发和维护，性能比较好，相对软件负载均衡稳定可靠些。缺点是：费用昂贵，难以拓展功能和容量，灵活性差。

软件负载均衡

LVS（Linux Virtual Server）跑在网络层的第四层，TCP或者UDP这一层，它是开源的，已集成在Linux内核中，其可伸缩，可以弹性部署，非常可靠。

Nginx跑在7层网络上面，它是轻量级的Web服务器，优势在于它有很好的网络适应性，只要后端的路由器可以连通就可以通过Nginx做HTTP的负载均衡，它支持URL、正则表达式等高级逻辑，同样是开源的。

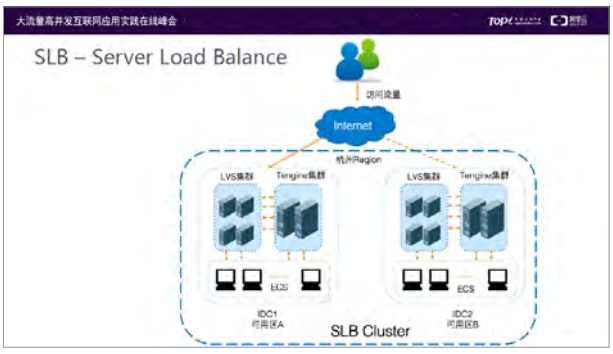

图3

SLB其实就是基于前面介绍的LVS和Nginx来实现的，分为四层负载均衡和七层负载均衡。如图所示，访问流量会经过公网的入口，进入某个可用区，每个可用区又会有多个机房，每个机房又会有一个LVS的集群和一个Tengine的集群来实现四层负载均衡和七层负载均衡。

图4

负载均衡技术已经作为了全集团的流量入口。第一，它是弹性计算的流量入口，服务阿里云的公有云用户，涵盖了大中小型网站、游戏客户、APP服务端，还有专有云，包括金融和政府部门。第二个服务的对象是云产品，比如RDS、OSS、高防等都用到了SLB的负载均衡技术，提供云计算服务的流量入口。第三，为集团VIP统一接入平台提供负载均衡服务作为电商平台的流量入口。第四，蚂蚁金服使用负载均衡服务作为支付宝、网上银行交易平台的流量入口。

高性能的负载均衡

图5

如何对负载均衡进行高性能的优化？首先是FULLNAT技术，它是LVS的一个转发模式，此优化的目的是为了摆脱开源LVS对网络部署的限制；另外的一个优化点是二层转发，LVS是Linux内核里面的一个模块，需要经过Linux传统的协议栈，性能非常低，二层转发可以通过记录MAC地址绕过Linux路由表，提升性能；第三个比较大的优化是CPU的并行化，可以让每个 CPU 并行的处理报文的转发；第四个优化是FASTPATH，完全旁路了Linux协议栈，直接将报文送到网卡，性能达到硬件线速；第五个优化是CPU指令的优化，由于FULLNAT需要对报文做一些修改，我们使用CPU专门对Crc32定制的计算指令会大大优

化计算校验的性能；最后一个优化点充分利用了现在服务器NUMA的特性，NUMA是计算机的一种架构，在不同的CPU上面访问特定内存或者系统资源会比较快，通过NUMA的特性来分配本地内存可以达到很大的性能提升。

FULLNAT优化

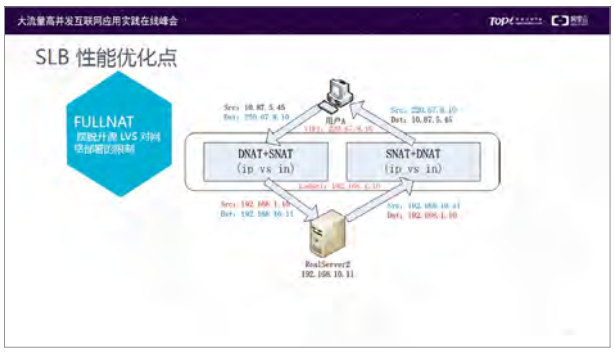

图6

CPU并行化

控制管理的并行化，修改了LVS的源码，原生的LVS本地只有一份配置是全局的，每个CPU访问它都要去加锁，我们的优化是在每个CPU上都有全量的转发配置的部署，这样，每个CPU查询自己的资源的时候是不需要上锁的，大大提高了处理的效率。每个 CPU 在查询连接（session）时，也是并行处理，大大提高了转发性能。

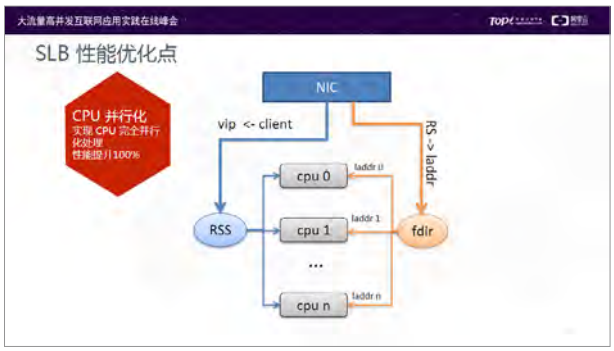

图7

当CPU分开工作时，怎么保证同一条流的入方向和出方向都落在同一个CPU上呢？之前提到的FULLNAT有两条流，一条是client访问vip，另一条是从RS访问local地址，这两条流采用了硬件的两个特性来保证他们落在同一个CPU上，client -> vip这条流采用了硬件的RSS技术，通过哈希自动选择CPU，通过网卡队列CPU中断的绑定使得关系固定下来，当它处理完发送到后端后，RS -> localaddr回来的入包的源地址是后端的服务器，目的地址是本地地址，此时通过 flow-director 特性对本地地址设置规则来保证分流的效果，即入包和出包的规则一致， 命中同一个 CPU 进行处理。

FASTPATH优化

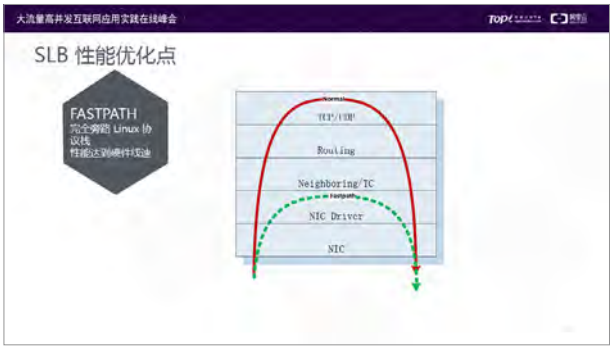

图8

只经过网卡层和驱动层，大大提升报文的转发性能。

高可用的负载均衡

图9

连接级的高可用通过Session同步来实现，LVS上面每一条转发的连接都会用Session来记录原地址、目的地址等相关信息，每台LVS上面都可以获得其他LVS上面的Session，这样当一台LVS宕机之后，其流量就会分配到其他LVS上面；第二个优化是单机高可用，每台LVS有两个物理网口，双上联了一台交换机，从网络路径上提供了一个冗余，每台交换机上面的路由器也有vip优先级，当一台路由器故障时它会根据优先级切换到另外一台路由器上，实现单机高可用；第三个优化是集群高可用，即热升级无感知、单机故障无感知、交换机故障无感知；第四个优化是通过主备双SITE、秒级切换实现AZ高可用；第五个优化是安全防护，使用SynProxy防御快速校验TCP的三次握手抵御攻击，减少CPU和内存的损耗；最后一个实现高可用的途径是健康检查，及时剔除异常RS，保证服务高可用。

负载均衡集群的高可用

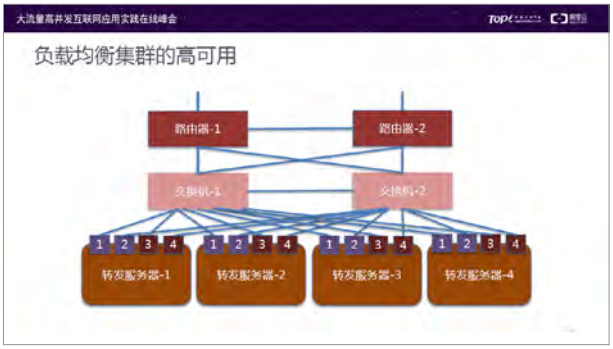

图10

每个集群可能有4台或者8台LVS作为转发服务器，每台服务器上面会有双网口，分别上联两台交换机，两台交换机又分别上联两台路由器，这样网络路径上的冗余就变得非常大，任意一条路径断掉不影响其性能。

负载均衡AZ的高可用

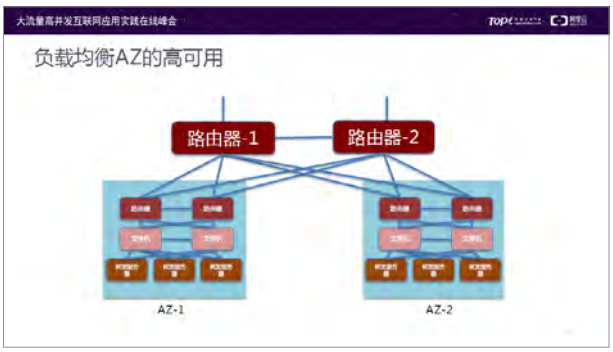

图11

以杭州集群为例，分为可用区A和可用区B，其上面都有全量的转发规则，我们可以设置某个vip，它的流量只通过路由器1来访问其中的某个集群。当整个的可用区发生故障时，它会秒级的切换到另一个路由器上，它的请求就会落在备用的可用区上面，这样当集群发生大规模故障的时候，vip的可用性不会受到影响。

SLB可用性全景图

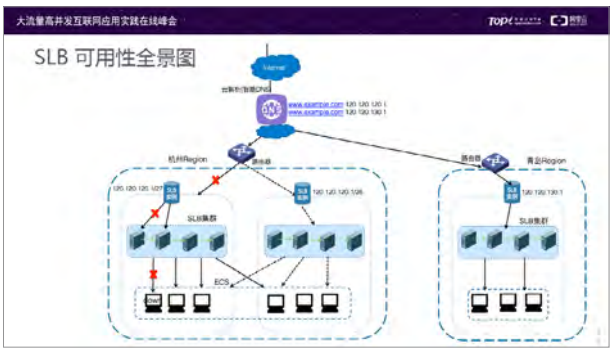

图12

首先可以通过DNS这种传统的模式在一个域名下面挂两条以上的vip，其中一条vip放到杭州集群，一条vip放到青岛集群，进到杭州集群的流量也有两个可用区，正常情况下会落到某个可用区上面，而另外的可用区是一个备份，当某个可用区变得不可用时，会在杭州集群内进行切换，实现了AZ的高可用。某一个可用区里面，某一台LVS宕机时，那么其他三台LVS就会提供服务。整体上实现异地多活，同城多活的架构。

展望未来

图13

1.软件架构：从内核态切换到DPDK架构，转移到用户态做网卡的轮询，报文的转发。

2.硬件升级：从10G网卡升级到40G网卡，集中硬件资源、缩小集群的数量，配备更快的CPU、更大的内存。

3.高可用：计划做跨AZ的Session同步，即使AZ之间的切换也让用户无感知，保证原有连接的持续；网络的全链路监控包括区域链路上报文转发的时延、丢包率、重传率等给客户提供更全面的网络质量评估；vip的主动探测帮助客户监控vip是否可用，并且把数据展示给用户；SLA，关于用户使用资源上的限制和保证，做好公有云用户的资源隔离。

专题聚焦

大流量高并发互联网应用实践在线峰会

专题简介：大流量、高并发一直都是技术热点，本次峰会邀请10位淘宝开放平台和阿里云技术大V，为广大开发者分享阿里的第一手实践，涉及海量订单时实同步与处理 、大数据驱动的客户运营 、聚石塔容器技术实践、千牛旺旺的云到端、探索大数据开放处理平台之路、无线开放生态内容、RDS数据库、MaxCompute、中间件、负载均衡等多方面内容。

本主题链接：
https://yq.aliyun.com/topic/64

云计算时代，负载均衡如何优化才能让性能起飞?

高人自有妙计：罗龙九六招制服云数据库大流量峰值

阿里无线开放新生态

10年老兵带你看尽MaxCompute大数据运算挑战与实践

不是你不会做菜，你只是缺个好厨房：深谈御膳房架构演进

订单同步有技巧，双十一高峰不再怕

千牛挂“虹（Rainbow）”，域和角色不胜数

Level up!从流量经营到客户运营实战技术分享

遵循互联网架构“八荣八耻”，解析EWS高质量架构6个维度的20个能力

鹰眼跟踪、限流降级，EDAS的微服务解决之道

数据智能时代，语音交互将是第一爆发领域

作者：初敏

摘要： 从语音识别与合成、人机对话、应用案例分析等方面分析为何一个数据驱动的智能时代，语音交互会是第一爆发领域，并将形成新一轮入口之争。

自从谷歌的Alphago战胜李世石后，人工智能在全世界范围内引起了高度关注。细看近年来备受热议的人工智能案例，实际上是机器学习特别是深度学习技术的发展和普及的结果。而今天的深度学习，跟三四十年前神经网络技术在原理上其实没有本质差别，最大的差异就是网络规模。以前大家只敢尝试一个隐含层，今天语音识别中常用的是7、8个隐层，甚至有人尝试一百多个隐层。以前一个隐层上也就放二三十个节点，今天可以放1024或2048个。我们之所以可以这么任性地增加网络规模，并不断构建出各种复杂的网络结构，一方面是计算能力的增强，另一方面是可以用来训练模型的数据规模的增加。因此可以说，近几年人工智能发展最主要是大数据驱动的机器学习技术的发展。

今天我们所做的学习，其实是在向数据学习；而今天看到的机器智能，大多数是从数据中学来的。因此，现在是一个数据驱动的智能时代。

图1：阿里云数据智能图谱

阿里在这个方向上，做了大量的布局，比如文字识别、人脸识别、图象识别，特别在电商领域做很多图象的分析。

我们为什么称之智能语音呢？这是因为语音不仅仅局限语音识别本身，同时还包括对所得到的文字的真正理解，甚至进一步的交互，这样才具有真正的智能性，而并非传统的将语音转化为文字。语音在人工智能这个圈子里，可以说是最成熟、最接近应用的领域之一。随着移动互联网时代的到来，手机、智能家居等设备呈现小型化、无屏化的趋势，语音就成为了一个最方便的入口。因此，在这个正在到来的数据智能时代，我们认为语音交互将是第一爆发的领域，将会形成新一轮入口之争。

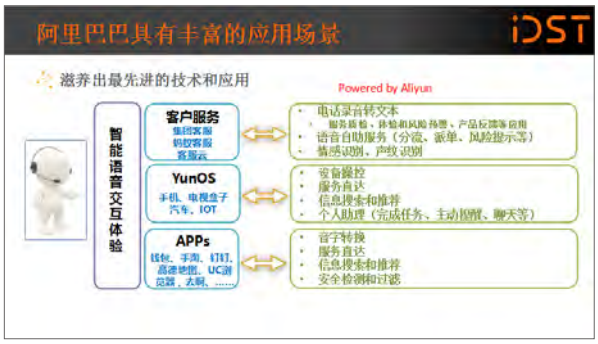

图2：阿里巴巴丰富的应用场景

到目前为止，阿里对语音的研发大概只有一年多的时间。阿里本身具有很大的客服系统，每天都有几千个坐席用于电话服务，同时还保留通话录音。但是这些录音是无用的数据，因为没有人来听它，除了客服团队会对很小一部分进行服务质量的抽检调查。而客户为同一件事再打客服电话时，遇到一个新的服务人员，就又需要重复之前所讲过的事情，导致客户体验非常之差。

那么智能语音可在其中发挥怎样的作用呢？它能将这些录音转化为文字，再通过自然语言的处理加以应用。例如在“质检”场景中，从文字提取有用信息，检测每一通电话是否存在问题。以蚂蚁客服为例，原本30多人的质检团队只能抽检1%的通话。而使用语音智能质检系统后，只保留10+人的质检团队就做到了100%的质检。

语音识别与合成

上述讲的是目前人工智能整体的大背景，未来所谓的人工智能最核心的是数据驱动的人工智能。在整个过程中，不仅仅是一个算法、深度学习，其中最本质的是要用数据将其驱动起来，才能获得真正的智能。

我们目前所做的工作，主要集中在语音和人机交互两个方面，一部分是基础的语音识别、合成；另一部分是人机间的交互对话。首先介绍的是我们在语音识别方面的工作。

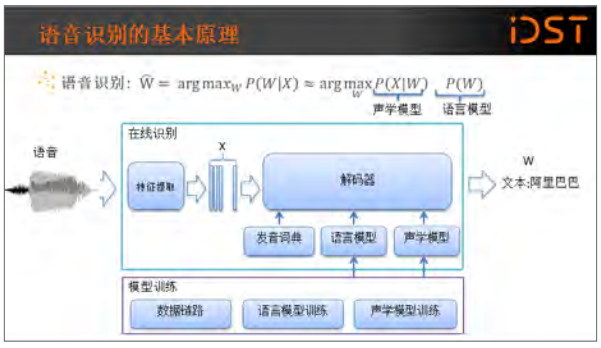

图3：语音识别的基本原理

如果将语音识别系统看成一个黑箱过程的话，那么它的作用就是把语音转换为文字的过程。从大体的原理上来讲的话，语音识别解码器最大后验决策的过程，给出一个语音的特征序列X,找出后验概率最大的一个文本串W。实际实施的时候，通过贝叶斯公式的分解成为两个模型，一个是声学模型，它的功能就是评估你的发音是什么，比如是发的是 b/p/m/f，还是d/t/n/l。 目前是使用深度神经网模型来完成；另一个是语言模型，这一部分则是评估哪一个文字串是更自然的语言。一般是用ngram模型，目前大家也在探索各种深度学习模型。另外还会用到发音词典这个资源。

其中获取声学模型和语言模型的过程称为模型训练过程。执行最大后验概率决策的过程成为语音识别解码过程。

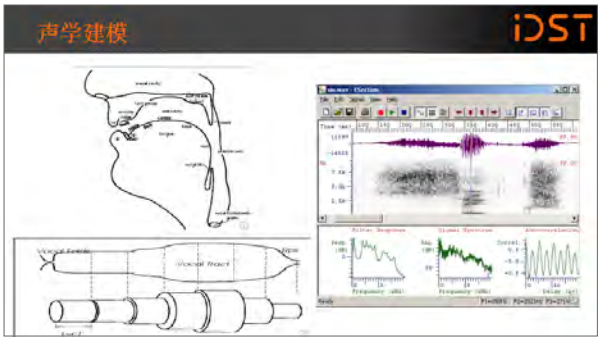

图4：声学建模

人的发音实际上是声带振动，通过振动产生周期性的波；声道相当于一个截面积不断变化的管子，不同形状的管子具有不同的共振频率,我们称之为共振峰，共振峰不同所发出的音就不同。 所谓的声学模型就是基于这类特征进行建模，比如说/a/和/i/的共振峰差异就很明显。最小的建模单位称之为音素（/a/、/i/、/u/、/z/、/c/、/sh/等）。在中文和英文中，最小单位是不同的，中文通常会大一些。

传统上比较流行的建模方式是采用马尔科夫链来描述一个音，包含不同的状态。但经过二三十年的发展，已经达到了尽头，每次优化的效果错误率下降仅仅相对8%-10%左右。在2011年，微软邓力、余栋等在大规模连续语音识别任务上成功的应用的DNN深度学习模型。它是把这个语谱图灌进去，在马尔科夫链的基础上，再用深度学习训练，可实现30%相对错误率的下降。在此基础上，语音识别就逐步变得可用起来，因此可以说深度学习最初的成功是在语音识别方面的，这是因为语音识别是一个非常好的封闭学习系统，学习目标是非常清楚的。

刚才所讲的是一个简单的DNN的模型。随着深度学习的发展，人们逐步在模型的拓扑结构上做文章。LSTM是一个RNN模型，通过设置门的开关有选择地实现记忆与遗忘。另外一

种是BLSTM，其在进行当前判决时不仅考虑历史数据，还会等待后面的数据进来后一起用来做判决。所以准确率会大大提高，相比于DNN模型，又可以实现错误率25%左右的相对下降。但是它带来的问题是：因为要在收到右边的内容后才能完成现在的判决，在时间上，就会形成判决的延迟。因此我们目前做的是长度受限（LC）BLSTM，兼顾准确性和时效性。该模型计算复杂度比较高，应用的难度在于时效性。我们在这个方面做了很多优化工作，最终使得这个算法可以达到0.6倍的实时，并完成第一个工业界生产系统的部署。如今，这个系统已经成为阿里云云栖大会的标配（提供实时语音字幕）。

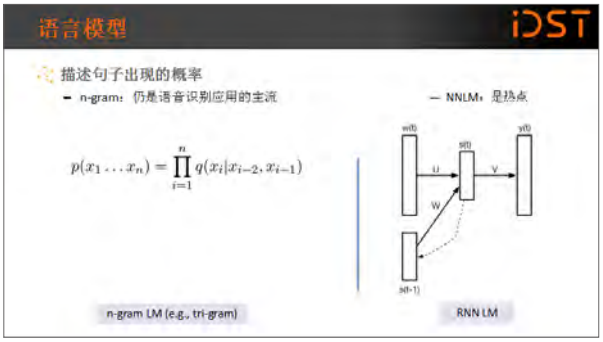

图5：语言模型

关于语言模型，它本质就是描述句子出现的概率。通常符合人说话规律的句子的概率会高于随机词组组合而成的语句。过去流行的模型是n-gram模型，现在仍然是主流模型之一。但是目前的研究热点是RNN模型。从套路上讲，语音识别在过去的二三十年内并没有发生太大的变化。真正的变化在于深度学习本身。

图6：数据规模和计算效率至关重要

在今天会议现场，大家可能注意到在我讲话的时候，可以实时产生滚动字幕,这就是我们的小Ai语音识别系统。小Ai的这项能力，今年3月首次在内部亮相，当时小Ai参加了阿里云年会，并当场跟中国速记第一人姜毅进行了PK。最终，小Ai以微弱优势胜出。

我们是春节前接到要在阿里云年会上进行人机PK的任务，包含春节假期一共不到一个月的准备时间。为了取得最好的效果，我们决定采用BLSTM模型，是深度学习中一个比较复杂但学习能力更强的模型。这个模型当时还在研发阶段。所以大家兵分两路：一路同

学利用我们已经采集到的1万多小时手机语音，做各种实验，来确定模型的最佳结构和参数,这就意味在数十块GPU卡上，并行进行好几组实验。两三周的时间这个模型小组完成了几十组对比实验。与此同时，另一路同学在集团内外到处收集各种演讲数据，在网上收集关于云计算、大数据领域的各种新闻和文章。这些数据的目的是帮助小AI适应垂直领域演讲。

刚才讲的是语音识别大的框架，如果说难是非常难。因为必须把每一个细节都十分完美地解决，最后才能得到特别好的效果。但整体来看，并没有特别神奇的点，仅仅是在不同的深度学习的模型上进行调试，重要的地方就是迭代能力和数据量的大小。因此数据的采集和使用就变得尤为重要，所以机器学习远远不是只研究某个算法，对企业而言，真正好用的数据模型一定是经过大量的数据验证的。

对于语音合成的前端处理，之前比较流行的是用CRF算法来预测停顿边界和等级，现在大家更多的尝试使用机器学习来解决这个问题。声音合成部分目前存在两种方法,一种是参数合成；另一种是波形拼接合成。

人机对话

刚才所讲的是语音的识别与合成，但这相对于今天所说的智能语音而言是远远不够的，这是因为我们希望在识别过程中能够进行理解，可以进行人机对话、交互。

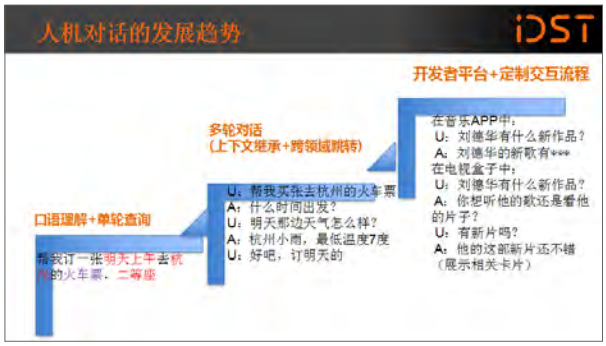

图7：人机对话的发展趋势

从上图可以看到，人机对话分为口语理解+单轮查询、多轮对话、开发者平台+定制交互流程三个阶段。其中各阶段最为核心的在于自然语言的理解，例如在“订一张上海飞北京的头等舱，下午5点出发，国航的”语句中，通过分类器将场景中最为重要的参数提取出来，然后用到火车票的数据服务去取结果并返回给用户。但用户往往不能在一句话中把所有的信息都提供出来。那么就需要通过多轮对话明确用户意图，一般是分为两个阶段：第一阶段，通过对话得到结构化查询；第二阶段，将查询的结果通过自然语言反馈给用户。

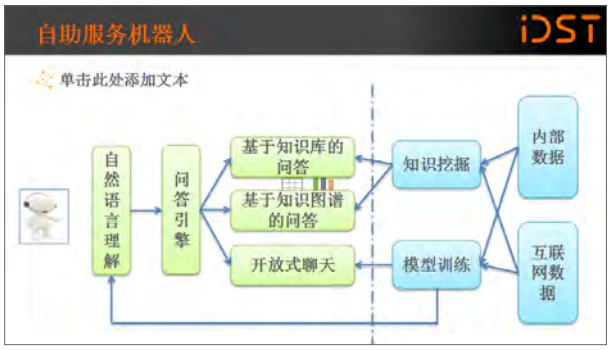

图8：自助服务机器人

在问答场景中，需要准确找到用户问题的对应答案。通过问答引擎后，又分为三种形式：基于知识库的问答、基于知识图谱的问答、开放式聊天。每个企业都肯能用户自己的FAQ或者知识库或知识图谱，数据来源可以是企业内部数据库或互联网数据。

图9：赋能生态圈

刚才所讲的这些技术点，阿里目前也正在做。我们希望能够自行搭建最核心的基础平台，然后提供给开发者用于定制化开发。所以我们会做底层核心技术的研发，在此之上提供了一些定制工具。通过用户上传数据或者典型的资料，对应的在用户所处的环境内进行优化。

在客户端，因为语音是比较复杂的，因为它必须有个数据采集端（录音口），这一点尤为重要，如果录音出了差错，那之后的工作基本就等于白费了。因此一般选用麦克风矩阵进行采样，在噪声较大的环境中还需要降噪处理，以保证录音的质量。

今天我们通过阿里云数加平台发布了一部分成果，包含技术文档、SDK等等，感兴趣的听众可以去自行查看。

应用案例分析

刚才更多讲的是技术，下面我分享几个具体的案例。

图10：语音识别助力行业变革

我们和蚂蚁客服有着深度合作。在双11当天大概有500万用户的查询，实际上94%都是自动解决的，只有6%是通过人工解决。这背后采用了大量的人工智能的技术，如上图显示的“安娜”。这是一个自动问答机器人，不仅可以回答你询问的问题，而且会根据你的历史行为进行提早预测你可能遇到的问题并给出建议。

另外一个工作就是：在客服电话时，用户可以通过语音来表述自己的问题，通过智能语音识别和交互转接到对应的客服上，免去了传统的不停跟随提示按键的步骤，缩短了服务过程。

图11：YunOS手机中的个人助理

另外一个是在YunOS手机中的个人助理，其中包含了二十几个领域的信息，还包括一些可执行命令，例如设闹钟、发短信、打电话等。后续还会加入人性化的功能。

图12：阿里小蜜

最后一个案例是阿里集团客服的合作——手机淘宝中的阿里小蜜，它通过语音的交互实现售前、售中、售后的打通，全方位的为消费者服务。

总结

智能语音可以有很多创新的用法。在未来的几年内，智能语音一定会非常快地普及和推广开，并且应用于各类场景。

关于分享者

初敏博士,阿里云iDST总监

专题聚焦

云栖TechDay精华文章合集

专题介绍：云栖TechDay是阿里云的线下活动。每一期都有固定选题，力邀多位技术大咖深入分享。持续交付、Docker技术、智能语音、机器学习、大数据工业智能等，社区特别将云栖TechDay精彩文章整理成为合集，分享给大家。

本主题链接：
https://yq.aliyun.com/articles/66145

在线AI技术在搜索与推荐场景的应用

作者：徐盈辉

摘要：在2016双11中，深度学习和强化学习技术首次得到了大规模的应用。针对双11当天的大流量、高并发的场景，阿里创造性的提出了基于强化学习的实收搜索/推荐排序策略决策模型等技术。

阿里巴巴集团的研究员徐盈辉在分享《在线AI技术在搜索与推荐场景的应用》时，结合搜索和推荐场景详细介绍了电商搜索推荐的技术演变、阿里搜索推荐的新技术体系以及未来的发展方向。

电商搜索推荐技术演变过程

图1

对于阿里巴巴电子商务平台而言，它涉及到了买家、卖家和平台三方的利益，因此必须最大化提升消费者体验；最大化提升卖家和平台的收益。在消费者权益中，涉及到了一些人工智能可以发力的课题，如购物券和红包的发放，根据用户的购物意图合理地控制发放速率和中奖概率，更好地刺激消费和提升购物体验；对于搜索，人工智能主要用于流量的精细化匹配以及在给定需求下实现最佳的人货匹配，以实现购物路径效率最大化。经过几年的努力，阿里研发了一套基于个性化技术的动态市场划分/匹配技术。

图2

电商搜索和推荐的智能化演进路程可以划分为四个阶段：人工运营和非智能时代、机器学习时代、准人工智能时代、人工智能时代。人工运营和非智能时代，主要靠领域知识人工专业运营，平台的流量投放策略是基于简单的相关性+商品轮播；在机器学习时代，利用积累的大数据分析用户购物意图，最大化消费者在整个链路中可能感兴趣的商品；准人工智能时代，将大数据处理能力从批量处理升级到实时在线处理，有效地消除流量投放时的误区，有效地提高平台流量的探索能力；人工智能时代，平台不仅具有极强的学习能力，也需要具备一定的决策能力，真正地实现流量智能投放。

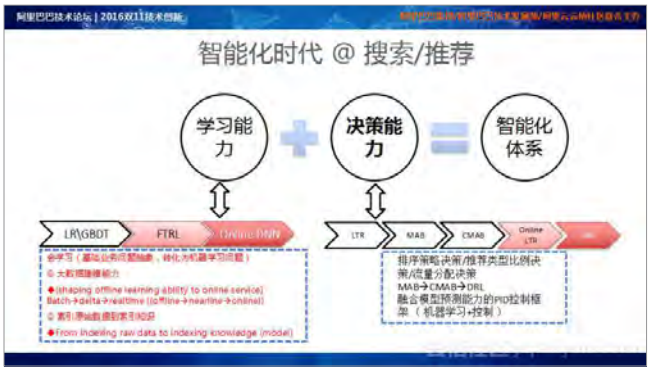

图3

智能化时代，对于搜索和推荐而言，可以提炼为两点：学习能力和决策能力。学习能力意味着搜索体系会学习、推荐平台具有很强的建模能力以及能够索引原始数据到索引知识提升，学习能力更多是捕捉样本特征空间与目标的相关性，最大化历史数据的效率。决策能力经历了从LTR到MAB再到CMAB再到DRL的演变过程，使得平台具备了学习能力和决策能力，形成了智能化体系。

借他山之石以攻玉

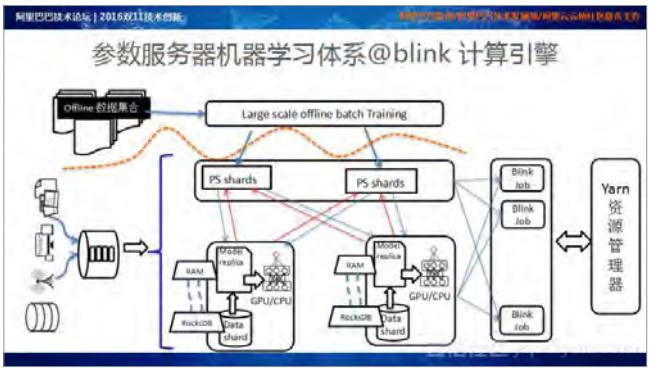

图4

借他山之石以攻玉。在线服务体系中，我们基于参数服务器构建了基于流式引擎的Training体系，该体系消费实时数据，进行Online Training；On Training的起点是基于离线的

Batch Training进行Pre-train和Fine Tuning；然后基于实时的流式数据进行Retraining；最终，实现模型捕捉实时数据的效果。

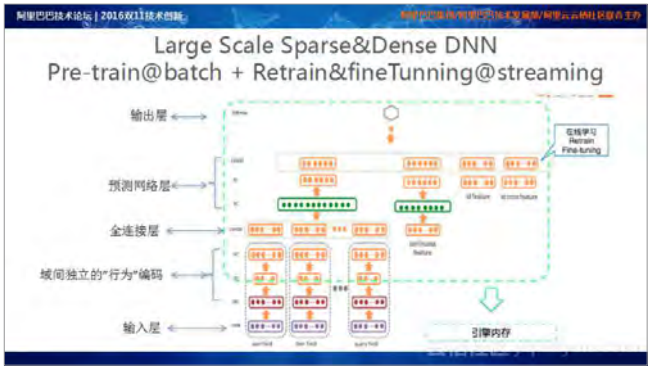

图5

上图是基于Wide & Deep Learning for Recommender Systems的工作建立的Large Scale Sparse&Dense DNN训练体系的架构，该架构中利用Batch Learning进行Pre-Train，再加上Online数据的Retrain&fine Tuning。模型在双11当天完成一天五百万次的模型更新，这些模型会实时输送到在线服务引擎，完成Online的Prediction。

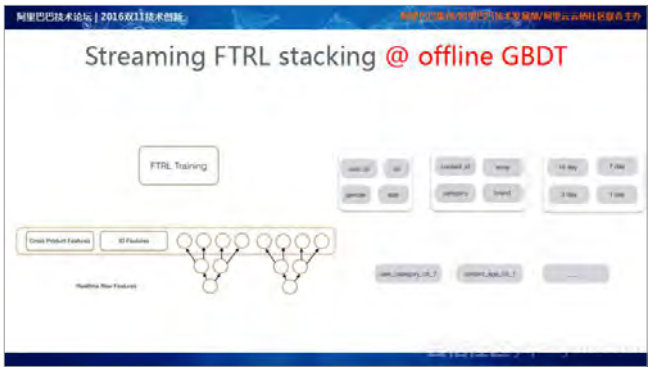

图6

Streaming FTRL stacking@offline GBDT的基本理念是通过离线的训练，在批量数据上建立GBDT的模型；在线的数据通过GBDT的预测，找到相应的叶子节点作为特征的输入，每一个特征的重要性由online training FTRL进行实时调整。

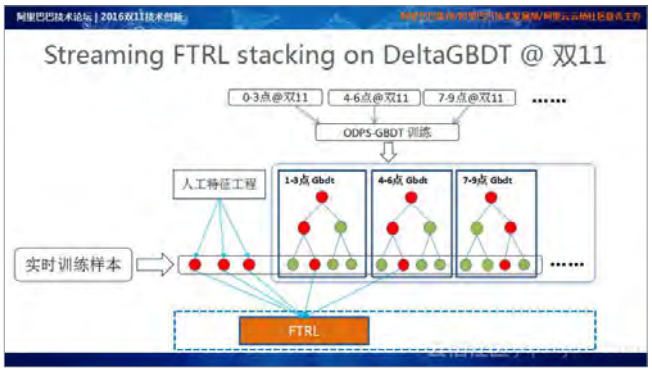

图7

双11当天的成交额是是普通成交日的十到十二倍，点击量将近三十倍。在用户行为密集发生的情况下，有理由相信数据分布在一天内发生了显著的变化，基于这样的考虑，GBDT的Training由原来的日级别升级到小时级别（每小时进行GBDT Training），这些Training的模型部署到Streaming的计算体系中，对于实时引入的训练样本做实时的预测来生成对应的中间节点，这些中间节点和人工的特征一起送入FTRL决出相应特征的重要性。

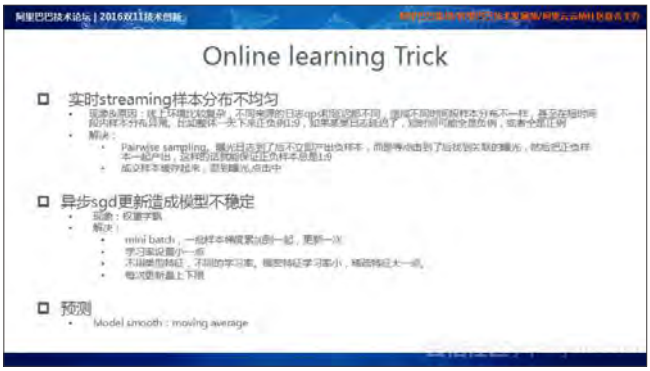

图8

Online Learning和Batch Learning有很大的区别，在Online Learning的研发过程中，总结了一些技巧：

◆实时streaming样本分布不均匀时，由于线上环境比较复杂，不同来源的日志qps和延迟都不同，造成不同时间段样本分布不一样，甚至在短时间段内样本分布异常。比如整体一天下来正负例1:9，如果某类日志延迟了，短时间可能全是负例，或者全是正例，很容易导致特征超出正常值范围。对应的解决方案是提出了一些 Pairwise sampling：曝光日志到了后不立即产出负样本，而是等点击到了后找到关联的曝光，然后把正负样本一起产出，这样的话就能保证正负样本总是1:9；成交样本缓存起来，正样本发放混到曝光点击中，慢慢将Training信号发放到样本空间中。

◆异步sgd更新造成模型不稳定时，由于训练过程采用的是异步SGD计算逻辑，其更新会导致模型不稳定，例如某些权重在更新时会超出预定范围。对应的解决方案是采用mini batch，一批样本梯度累加到一起，更新一次；同时将学习率设置小一点，不同类型特征有不同的学习率，稠密特征学习率小，稀疏特征学习率大一些；此外，对每个特征每次更新量上下限进行限制保护。

◆预测时，在参数服务器中进行Model Pulling，通过采用合理的Model smooth和Model moving average策略来保证模型的稳定性。

智能化体系中的决策环节

电商平台下的大数据是源自于平台的投放策略和商家的行业活动，这些数据的背后存在很强Bias信息。所有的学习手段都是通过日志数据发现样本空间的特征和目标之间的相

关性；进而生成模型；之后利用模型预测线上的点击率或转化率，由于预测模型用于未来流量投放中，因此两者之间存在一定的时间滞后（systematic bias），也就观测到的数据和实际失效的数据存在着Gap。在工作逻辑中，如果一个特征和目标存在很强的Correlation，则该特征就应该在线上的预测中起到重要作用。

图9

在整个体系中，Systematic Bias很难做到理想化的控制，而且离线模型预期效果与线上的实际生效效果存在差异，这背后的根本原因是Correlation并不等于Causation，也就是特征与目标相关并不意味着特征出现一定导致目标发生。

那么怎么解决Offline Reward Signal不等于Online Dashboard Metrics的问题呢？我们引入了强化学习，通过引入Online User Feedback更好地定义Reward，对线上排序策略进行调整，使其具有更强的自适应性。

搜索/推荐引擎决策体系

图10

搜索引擎和投放页面天然存在互动：搜索引擎观测消费者的交互状态；搜索引擎根据交互状态执行投放策略；投放策略之后，呈现商品结果页，消费者在商品结果页中的操作行为反馈给搜索引擎。引擎决策实际上能改变投放环境，进而影响消费者的交互，改变消费者

的状态。如果不需要建立从状态到动作的策略映射，可以采用Multi-armed Bandits方法进行流量探索；如果需要建立该映射时，需要采用Contextual MAB方法；在新状态下，考虑消费者的滞后Feedback对于引擎在之前状态下的Action正确与否产生影响，需要引入强化学习的思想。

搜索和推荐过程可以抽象成一个序列决策问题，从消费者与引擎的交互过程中寻找每一个不同状态下的最优排序策略（各种排序因子的合理组合）。

图11

我们的目标是希望搜索引擎决策体系进化为具有强化学习能力的智能化平台。过去的搜索，我们只能做到遇到同样的用户购物诉求下，尽可能保证做得不必以前最好的方法差，也就是所谓的Historical Signal==Best Strategy；一切模型都是建立在优化直接收益的基础上。未来的搜索，我们希望能够保证长期收益最大化来决定引擎的排序策略，也就是Immediate Reward+Future Expectation=Best Strategy；未来的排序融合入模式都是建立在优化马尔科夫决策过的基础上，最大化The Discounted Reward。

基于强化学习的实时搜索排序调控

下面简要介绍下为应对今年双11提出的基于强化学习的实时搜索排序调控算法。

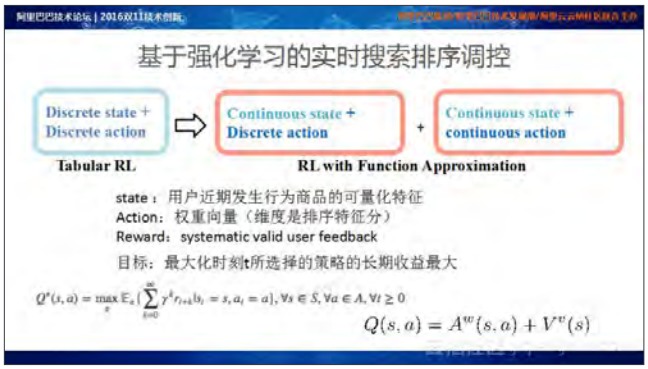

图12

对于强化学习，它的目标是最大化时刻T所选择的策略的长期收益最大。对于离散state和离散Action的情况，可以采用Tabular RL方法求解；对于连续State和连续Action，

采用RL with Function Approximation。其中State表示用户近期发生行为商品的可量化特征，Action表示权重量化（维度是排序特征分），Reward是Systematic Valid User Feedback。

双11采用Q-learning的方式进行实时策略排序的学习，将状态值函数从状态和策略空间将其参数化，映射到状态值函数的参数空间中，在参数空间中利用Policies Gradient进行求解；将状态值函数Q拆解成状态值函数V(s)和优势函数A(s,a)进行表达。

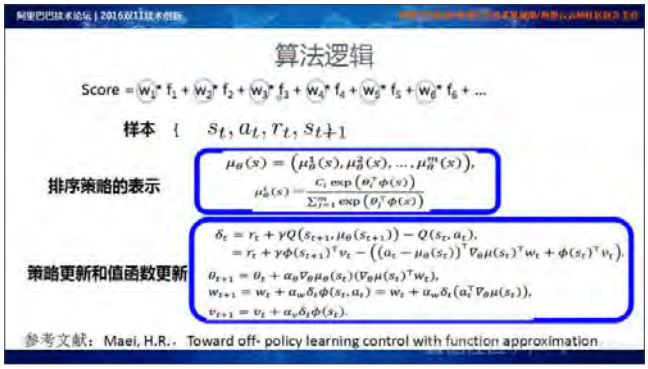

图13

其算法逻辑如上图所示，基本算法是实现线上几十个排序分的有效组合，样本包括日志搜集到的状态空间、Action Space（这里对应的是排序分空间），奖赏是用户有效的Feedback，具体的排序策略表达公式以及策略更新和值函数更新的公式可以参考Maei，HR的《Toward off-policy learning control with function approximation》一文。

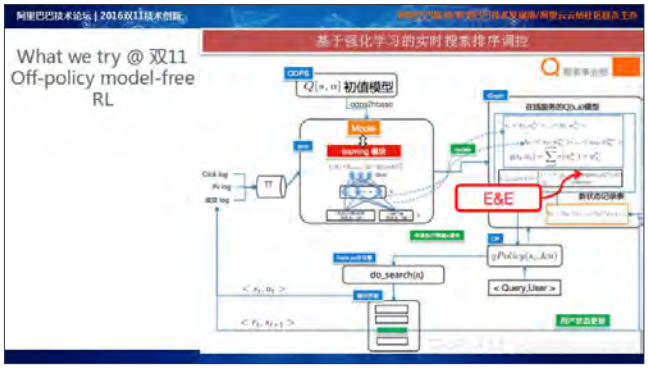

图14

在双11采用的基于强化学习的实时搜索排序调控的实现体系如上图所示。当用户输入query时，会向系统询问哪一种排序策略最适合自己；该查询策略请求会上传至在线策略决策引擎，在线策略决策引擎通过实时学习的Q(s，a)模型合理选择有效策略，然后再返回给搜索引擎；搜索引擎依据当前状态下最有效策略执行搜索排序；在搜索排序页面展示的同时，系统会及时搜集相应的状态 action以及用户feedback的信号，并进入到Online Training Process；而Online Training Process会通过Off-policy model-free RL方法学习State To Action的映射关系，再从映射关系中得到线上排序所需要的策略参数；该策略参数由在

线策略决策引擎通过Policy Invalid Process输出给在线搜索引擎。

总结

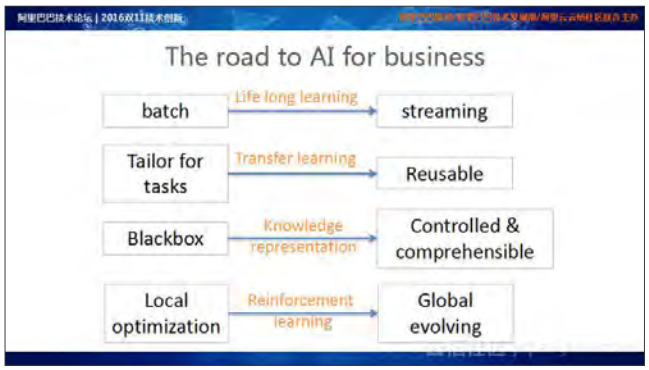

图15

整体搜索/推荐希望建立一个Close-loop for iCube learning体系，其中iCube要求系统具备immediate、interactive、intelligent的能力。整体从日志搜集到maximize rewards、minimize dynamic regret实现Online Training；其中Training模块能够高效地部署到Online Service；而Online Service必须具有很强的探索和overcome bias能力，进而使得整个体系能够适应新的数据，提升流量投放效率，同时能够探索新奇和未知的空间。

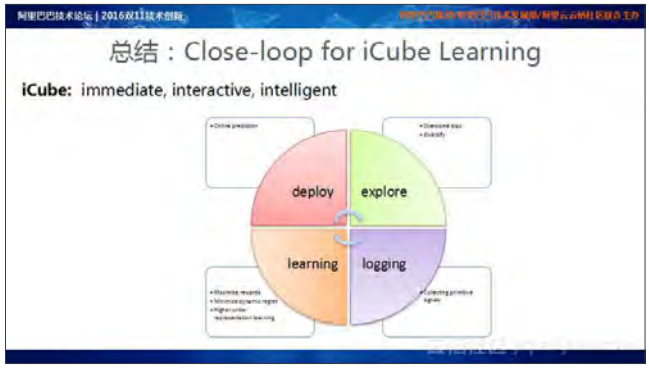

图16

在AI应用到商业的过程中，未来努力方向是：

◆From batch to streaming，希望从historical batch learning转化为life long learning；

◆整个学习体系由tailor for tasks 向利用transfer learning实现不同渠道、应用下学习模型的复用转变；

◆Training process 从Blackbox转变为实现合理的knowledge representation，实现线上投放逻辑的controlled&comprehensible；

◆学习体系随着强化学习和在线决策能力的增强，从local optimization向global evolving转变。

专题聚焦

2016双11技术创新

策划方向：每年的“双11”是阿里技术的大阅兵和创新能力的集中检阅。1207亿交易额的2016年的“双11”背后，更是蕴藏了异常丰富的技术实践与突破。从承载亿级用户大流量的网络自动化技术，到资源充分利用的超大规模Docker化；从支撑最大规模在线交易的实时和离线计算能力，到人工智能在搜索和推荐场景下的创新应用；从颠覆购物体验的VR互动，到背后千人千面的商铺个性化；从应对前端极限挑战的“秒开项目”，到绚烂媒体大屏背后全面的数据架构，这些璀璨的内容组成了2016第四场在线技术峰会。

本专题链接：
https://yq.aliyun.com/topic/74

本专题文章：

云栖社区技术团队名录

学习业内最新技术，分享实践创新成果，共建开放技术生态。已经入驻云栖社区的技术团队，都在这里，你可以找到更多志同道合的伙伴！

序号	名称	序号	名称
1	云角技术团队	43	购阿购大数据团队
2	短趣网络传媒团队	44	阿里云机器学习
3	阿里云存储服务	45	优云软件
4	阿里云数据库ApsaraDB	46	聚石塔AliAPM团队
5	阿里云数加---大数据开发	47	阿里云数据传输服务
6	阿里云云盾	48	阿里妈妈团队
7	阿里云 DataV 数据可视化	49	六次元AR
8	aliexpress-技术团队博客	50	神马搜索前端团队
9	阿里云容器服务(Docker)	51	阿里云数加QuickBI
10	阿里云资源编排服务	52	阿里云DataIDE
11	手机淘宝技术团队	53	阿里云服务
12	羿兔科技	54	阿里云研究中心
13	阿里云CDN	55	企业级分布式应用服务EDAS
14	阿里技术矩阵	56	阿里云监控服务
15	阿里云-iDST-智能语音交互	57	阿里云大数据孵化器团队
16	阿里云E-MapReduce	58	云市场头条
17	OneAPM	59	阿里云UED
18	阿里云持续交付平台	60	Alibaba Cloud Focus
19	阿里中间件团队	61	嘉为科技
20	阿里搜索事业部技术团队	62	菜鸟网络—物流云
21	驻云技术团队	63	阿里云数据市场
22	阿里巴巴客户体验驱动及创新中心	64	阿里云流计算
23	袋鼠云技术团队	65	阿里云 Serverless Computing
24	阿里云规则引擎	66	阿里云关系网络分析软件(Graph Analytics,简称I+)
25	阿里云网络产品	67	阿里云技术服务
26	阿里云移动数据分析	68	阿里百川
27	阿里高德AMAP应用技术团队	69	趣拍云团队
28	阿里云分析型数据库	70	阿里神盾局
29	阿里云移动服务	71	ApsaraSRE
30	阿里大于	72	阿里云弹性伸缩服务
31	API网关	73	Node地下铁
32	阿里云云服务器ECS	74	蚂蚁技术
33	db匠 ---- 阿里云数据库团队内核组	75	数据人社区
34	阿里云分布式存储	76	阿里云大学
35	阿里云数加团队	77	阿里双创在线
36	阿里云效平台	78	天猫技术团队
37	阿里人工智能&大数据	79	日志易
38	淘宝开放平台	80	飞猪旅行技术团队
39	阿里云大数据应用加速器DTBoost	81	钉钉开放平台
40	阿里千牛旺旺技术团队	82	阿里云内容安全
41	阿里聚安全	83	阿里数据库技术
42	阿里研究院		

以上按时间排序。欢迎更多朋友加入云栖社区！ 申请链接：https://yq.aliyun.com/teams