阿里云

企业物联网平台

IoT Platform

技术白皮书

文档版本：202203

01 行业趋势

1.1. 技术发展趋势

中国物联网设备连接数在2016~2020年间，年同比增长率为46.1%，经历了高速增长，预计到2025年会达到80亿。

阿里云物联网平台的核心价值是帮助企业设备数字化、智能化，设备产生的海量数据与企业的业务数据融合之后会产生巨大的价值，能够促进企业高效低成本地运营，进而提升整个社会生产效率。IoT在这波数字化浪潮中至关重要，让所有设备从孤立的变成有生命的。互联网时代和移动互联网时代的本质都是人的在线化，物联网时代百亿规模设备的在线化和数字化将会对物理世界进行重塑，尤其在5G网络的推动下会加速这个进程。

1.2. 面临的业务挑战

随着大量物联网场景开始涌现，海量碎片化设备和巨量时序数据给物联网平台带来了一系列新的要求和新的技术挑战。

- 高可用

  物联网从早期2016年主要应用在消费类智能家居场景，到最近几年场景越来越丰富，从文旅、园区、地产、城市、农业，再到工业、汽车等场景，其可靠性要求从民用级上升到了企业级。物联网平台的高可用能力决定了能够支撑客户业务持续运行的底线，且在应对大量影响民生安全、工业制造、社会稳定的场景时，需要提供极近苛刻的高可用能力。

- 性能

  物联网在互联网消息链路上新引入了一端（设备端），且应用端通过云平台到设备端的双向通信能力非常关键，设备状态的上报和呈现、设备指令的控制和执行，是物联网远程设备在线化、智能化的基线。随着场景的丰富，设备和应用间双向通信的RT、性能尤为关键，若指令延时过高，可能导致客户资损、民生安全受到威胁等意想不到的问题。

- 生态化

  物联网由于涉及到传统领域，链路长、角色多、终端多样性导致碎片化非常严重，因此，很难有一个角色或一家公司能够从头到尾将物联网升级全部完成。而物联网生态化趋势越来越明显，促进了全行业全面数字化升级，需要越来越多的角色进入到产业链。例如软件开发者、硬件开发者、模组商、芯片商、系统集成商、设备商等众多角色，需要物联网平台作为桥梁促进万物互通、标准化、以及生态化。

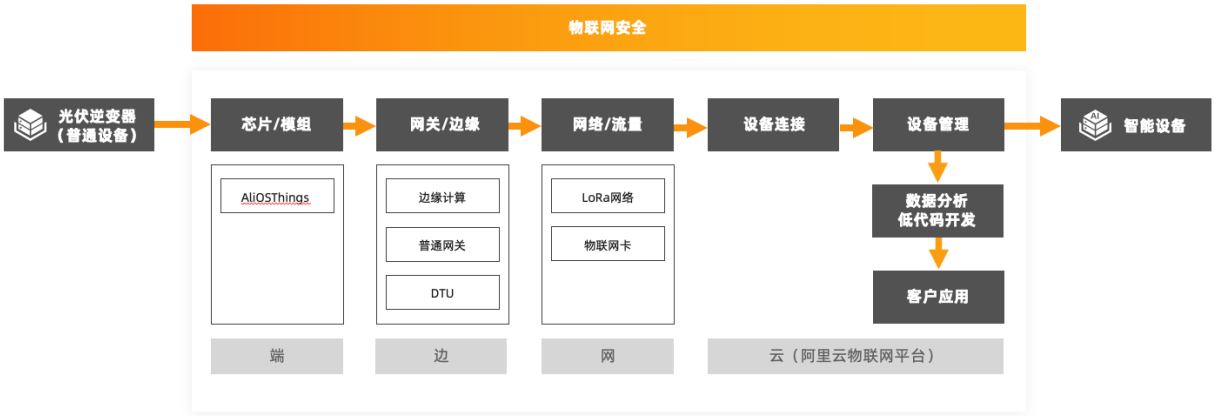
- 智能化

  所有场景数字化转型最终的目标是为了智能化，从而利用大量数据分析进行经营提效、降低成本、创新业务。物联网平台随着设备连接、管理、运维的发展，也开始逐步进入到数据智能的阶段。如果一台智能电表每隔15分钟采集一次数据，每天自动生成96条记录，那么全国接近5亿台智能电表，每天就能生成近500亿条记录。联网汽车、工业场景等设备上报数据会更频繁，据预测，五年之内物联网设备产生的数据将占世界数据总量的90%以上。超大规模数据为智能化带来了技术挑战，也带来了巨大的发展空间。

02 技术架构

2.1. 背景信息

阿里云IoT为企业数字化转型和设备智能化升级提供了一系列基础产品，一个普通设备升级为智能设备需要覆盖物联网端、边、网、云四大基础路径，解决设备通讯、计算、网络、连接、管理、数据、应用等关键问题。阿里云IoT在端、边、网、云上分别提供了相应的产品技术能力，包括AliOSThings操作系统、边缘计算、物联网卡及无线网络、物联网平台等。



2.2. 产品介绍

设备连接和管理服务属于物联网平台最基础的能力，帮助客户设备实现在线化、数字化，让客户不需关心物联网基础设施，完全聚焦在自己的核心业务上。

以光伏逆变器为例，如果客户自己要实现设备在线化、数字化，需要面临的问题有电站的采集器如何接入、采用什么数据传输协议、如何保障连接的安全和稳定；电表如何结构化建模、气象数据如何实时采集、逆变器故障如何预警、风机如何进行远程维护和固件升级；如何实现分销商累计发电量统计、电站故障率统计等等。以上问题无疑为客户智能化升级带来了接入门槛高、接入周期长、管理运维搭建难等问题。

一个新能源企业将光伏变电站搬上云

采集器接入	电站管理	智能发电
如何接入云端	气象数据实时采集	分销商累计发电统计
数据传输协议	逆变器故障预警	电站故障率统计
连接的安全和稳定	风机远程维护	同一城市平均发电量
	设备固件升级	

- **设备连接**

  设备的在线化，最核心技术在于设备连接和消息通信。一方面是物联网时代的设备连接，与互联网、移动互联网时代的PC、APP连接相比，有其特殊性，例如极度追求低功耗、低时延的资源受限设备；追求超高吞吐的海量点位场景；以及大量传统三方协议及行业协议业务。另一方面是消息量规模大，且可靠性、延时性、订阅灵活性与互联网面向人或应用的消息特点不太一样。

- **设备管理**

  设备的数字化，最核心技术在于设备建模和设备全生命周期管理。设备建模将设备投影到云上产生孪生体，设备孪生体和物理设备保持状态的一致性，并且能够实时双向通信，设备孪生体作为设备的抽象层，为上层应用屏蔽了物理设备的差异性。随着设备场景越来越丰富，对建模能力提出了非常高的要求。同时相较于互联网移动端，物联网设备存在地理位置广泛性、网络状况的不确定性、设备资源的差异性、高可用要求的严苛性、海量规模的高并发性等特殊性，为设备全生命周期管理带来了不一样的挑战，需要充分考虑可无人运维、大规模、数据异构、资源受限等因素。

以下会从设备连接和管理服务中，选择六个关键技术进行解读，包括安全稳定连接、海量消息、物模型（设备建模）、大规模设备管理、监控运维、异常检测。

03 技术优势

3.1. 安全稳定连接

3.1.1. 核心技术挑战

- **端侧碎片化**

  物联网场景覆盖的“物”种类非常广泛，必然导致碎片化，很难形成规模化效应和高附加值，而现在还没有产品能覆盖所有的场景，给物联网的平台企业带来很大的技术挑战。物联网平台接入层尝试用多样性的连接方式来解决不同设备的上云问题。

- **海量设备的连接稳定性**

  当并发连接数量达到千万甚至亿级时，对于任何一个平台而言，保持连接的稳定性都是很大的技术挑战。例如各种网络问题、时钟溢出导致的连接风暴、发布时导致的设备离线、设备端异常导致的死循环不停建连等，都是接入层需要解决的问题。

- **设备的网络质量**

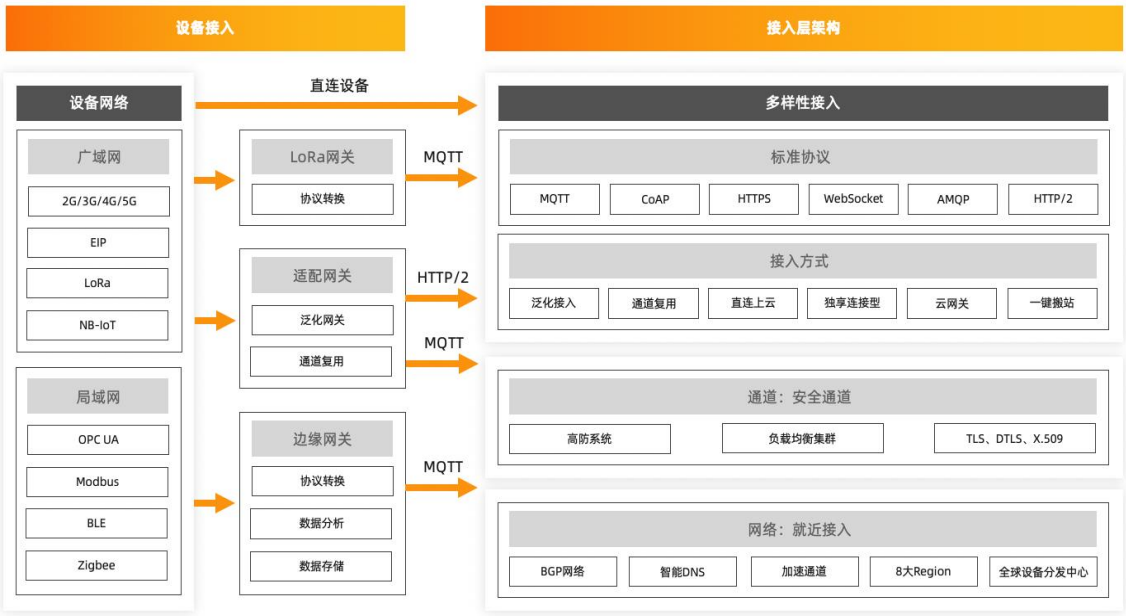
  设备种类广泛导致设备部署的位置不同，例如在地下、在高空、在荒野、在边缘地区或在海外等，怎么让不同设备都能有好的网络质量，是接入层首要解决的问题。因为设备连接上云是IoT的基础。

- **设备的安全性**

  各种设备都联网后，会给物联网的安全性带来更大的挑战，例如汽车、门锁、起搏器等受到安全攻击，都会对用户的隐私、财产、生命等造成严重的威胁。

3.1.2. 技术详细描述

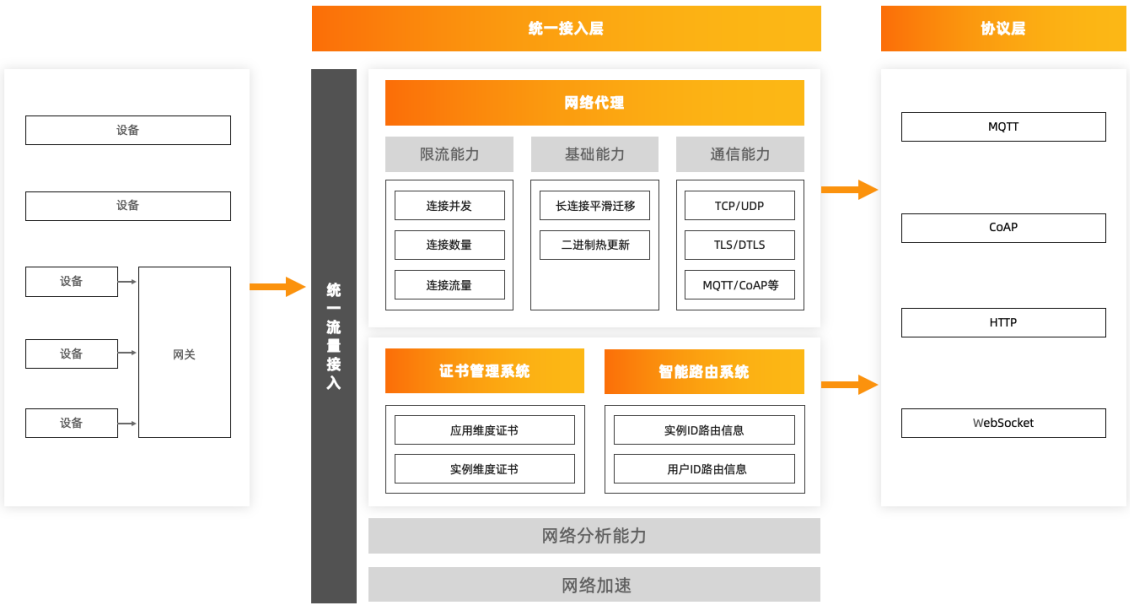
- 端侧碎片化：接入的多样性



针对碎片化的接入问题，通过不同的接入方式来适配不同的设备场景。

- 直连设备：对于广域网的设备，通过直连的方式直接上云。
- 局域网设备：通过不同的局域网协议接入边缘网关，有边缘网关转换协议、然后做数据处理，再通过MQTT协议接入物联网平台。
- LoRa设备：先接入LoRa网关，然后由LoRa网关通过MQTT协议接入阿里云物联网平台。
- NB-IoT设备：只能接入电信平台，阿里云物联网平台通过云云对接的方式，先对接电信平台，然后把设备接入到阿里云物联网平台。
- 私有平台的设备：通过泛化接入的方式，把私有协议转成标准的MQTT协议，然后接入到阿里云物联网平台。
- 通道复用：对于边缘网关，其下的子设备可以通过通道复用的方式上线和消息上下行通信，这类子设备与直连设备的能力对等。
- 云网关：针对采用了标准MQTT协议的设备，但自定义了设备身份信息和消息通信Topic的设备，通过云网关接入方式解决身份和Topic的标准化。

➢ 海量设备的连接稳定性



1) 连接限流能力

从外部请求限流和内部资源限流两个维度设计的接入层限流。针对外部限流，有并发建连限流，单连接流量限流，背压机制（结合业务层消费能力和TCP滑动窗口机制来实现），节流机制（溢出包丢弃）。针对内部资源限流，限制单进程、单应用的TCP Session数量，针对TCP缓存的内存限制，针对单应用的CPU使用限制。通过内、外资源的限流策略来防止连接层的雪崩，同时减少对下游系统的冲击。

2) 应用热更新能力

在网络代理发布时，会导致设备的TCP长连接断开，对于设备而言，需要重建连接，同时在建连的过程中消息不能到达。对于长连接断开，阿里云物联网平台支持了平滑迁移和缓慢下线的能力。通过老进程关闭listen fd，新进程接管listen fd，老进程维持24小时，让设备重连后自动迁移到新进程。对于长时间不重连的设备，通过缓慢下线的策略逐步使设备下线重连，减少同时大量设备下线对用户业务的影响。通过上述两个策略配合使用，可减少网络代理发布时对设备连接的影响。

3) Session转移

网络代理层跟协议层之间采用了TCP长连接，在协议层发布时会导致TCP长连接断开，在协议层保存了本地Session信息，如果当前发布机器的Session信息丢失，连接断开后需要设备重连才能恢复。针对上述情况，阿里云物联网平台设计了Session转移功能，在协议层发布时，可以把TCP长连接和设备的Session信息转移到其他未发布的机器上，此时协议层的发布可以做到对设备无感。

4) 快速容灾

为了解决单故障导致的全平台问题，阿里云物联网平台针对协议层、消息层部署了多个集群，同时会把不同的用户放到不同的集群里，当某个集群的协议层或者消息层出现系统异常的时候，网络代理层可以通过路由能力和Session转移能力把设备的连接转移到不同的集群，从而保障单集群故障能快速恢复。

- 设备的网络质量

设备网络是连接稳定性的保障，阿里云物联网平台为了让设备的网络质量更好，采用了全球分发技术，不论设备在哪里生产和注册，都可以在设备接入前把设备分发到离设备最近的地域，然后利用阿里云物联网平台的国内外8大地域部署的能力，让设备就近快速接入。同时为了解决不同地域设备快速获取就近接入点地址，物联网平台采用了全球加速的技术，在设备就近接入后，又采用BGP网络来解决地域内的网络质量问题。

- 设备的安全性

IoT平台基于四层安全设计和离线安全分析结合来保障设备的安全性。

1) 安全防御层

借助阿里云的DDoS、高防等能力，防止SYN洪水攻击等，做到流量的有效清洗，可以防止1000 Gbps以上的流量攻击。

2) 通道安全层

通过实现TLS、DTLS，X.509，ID²等安全加密技术，实现传输层的加密，防止数据在传输过程中被篡改、伪造等，同时针对低功耗设备，提供PSK、SessionTicket等能力，解决TLS过程中的数据传输量和网络RT的问题。

3) 身份安全

支持三种类型的设备身份，不同IoT场景可以使用不同的身份，保障每个设备都有唯一身份，同时对设备认证做了加签，防止身份的伪造。

4) 数据安全

按单元隔离不同集合的用户，然后再按用户维度和实例维度做更小粒度地隔离，保障数据在实例内、用户内、单元内闭环，做到每个用户只能看到自己的数据。

5) 离线数据分析

利用设备行为数据结合平台的AI能力，分析每个设备的安全性，针对安全等级低的设备做预警，并且跟平台安全层结合，针对攻击类设备实现自动拦截的能力。

3.1.3 核心技术点

技术	说明
安全能力	基于四层安全设计和离线 AI 分析能力，解决 IoT 平台的设备安全问题。
就近接入	通过全球设备分发能力和全球 8 大地域部署来支持设备的就近接入，做到国内地域内的设备接入的网络平均 RT 在 40ms 内。
多样性接入方式	支持 7 种标准接入协议、多种网关，满足不同网络类型、不同协议、不同功耗的设备接入。
Session 转移技术	支持百万长连接的设备 Session 在分钟内迁移，可用于容灾、发布断连等场景。
热更新能力	通过监听端口转移、24 小时新老进程长连接切换、缓慢下线三个技术手段来解决网络层发布带来的长连接平滑迁移问题。

3.2. 海量消息

3.2.1. 核心技术挑战

- 海量Topic

  传统的MQ如Kafka、RocketMQ等能支持的Topic数量非常有限，随着Topic数量的增加，吞吐量和性能会急剧下降。在IoT场景下，Topic数量是亿级别的，IoT消息队列必须支持海量的Topic。如何支持海量Topic，以及如何做到高性能、高可用，是IoT消息队列的核心挑战，也是最基础的能力。

- 实时优先

  传统的MQ如Kafka、RocketMQ等，若出现了消息堆积，则必须等堆积的消息消费完毕后，后续的消息才能正常消费，这是典型的FIFO模式。但是在IoT场景下，FIFO模式是不合适的。尤其是下行控制设备，实时性要求是非常高的，不可能等到堆积的消息都处理完后再把指令下发到设备。如何做到实时优先，同时又能正常处理堆积的消息，也是IoT消息队列的核心技术挑战。

- 规则引擎

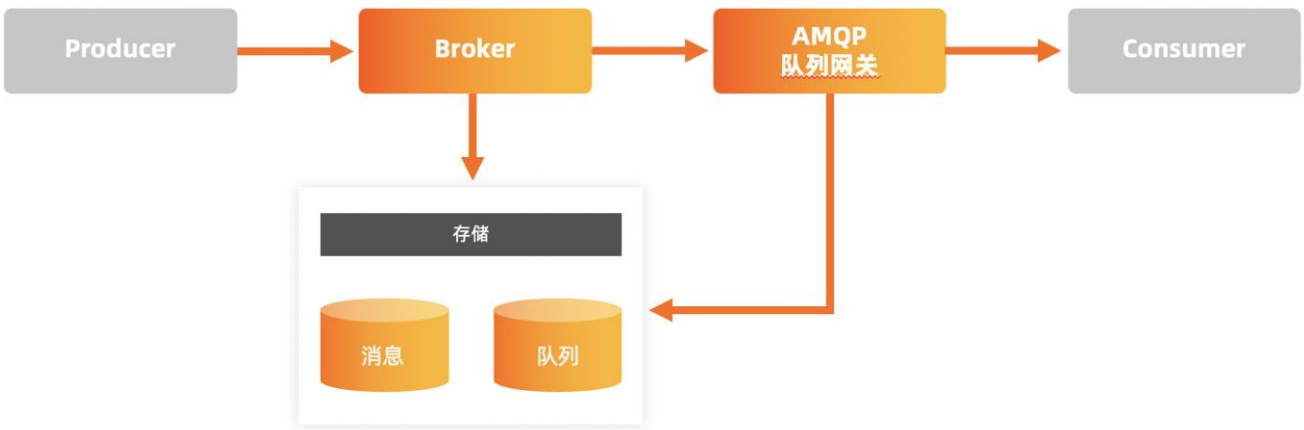
  与传统的MQ不同，传统的MQ往往聚焦于异步解耦、削峰填谷等。IoT消息队列的规则引擎是一个十分重要的功能。对接入平台的设备设定规则，在条件满足所设定的规则后，平台会触发相应的动作来满足用户的业务。典型业务有消息流转、场景联动等。面对各种复杂的业务场景，规则引擎如何设计、如何做到高性能、如何具备扩展性、如何保障稳定性，都是规则引擎的技术挑战。

- 高可用

  与传统的MQ相比，IoT消息队列在高可用方面的挑战更大。网络抖动、设备故障等外部因素，可能导致设备大规模的断网重连，进而造成消息流量洪峰。IoT消息队列如何应对亿级流量洪峰、亿级消息堆积；如何保障消息不丢；如何保障消息的实时性；如何保障高性能，都是很大的技术挑战。

3.2.2. 技术详细描述

- 海量Topic：计算存储分离



物联网平台消息队列支持架构升级优化，针对传统MQ的问题，结合IoT场景的业务特性，做了计算存储分离的架构。其中计算节点包括Broker和AMQP队列网关，存储采用NoSQL数据库。在这种架构下，各个节点职责清晰，可分别支持水平扩展。

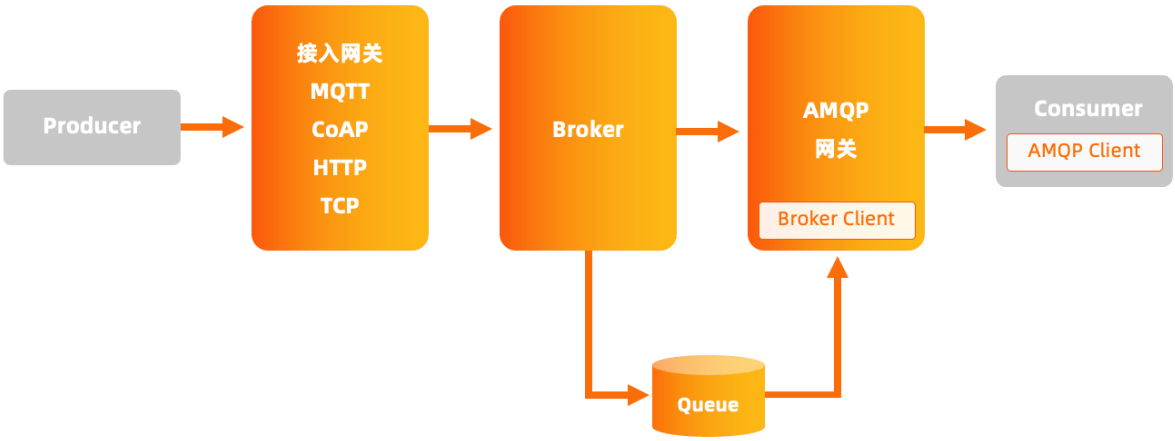
1) Broker是无状态的计算节点，职责明确只进行消息分发。

2) AMQP队列网关也是无状态的计算节点，负责消息推送以及离线消息拉取。

3) 消息存储采用NoSQL数据库，支持高吞吐读写。

通过以上计算存储分离的架构，解决了传统MQ无法支持海量Topic的问题。计算节点和存储节点都可以水平扩展，架构具有良好的弹性。

- 实时优先：推拉结合

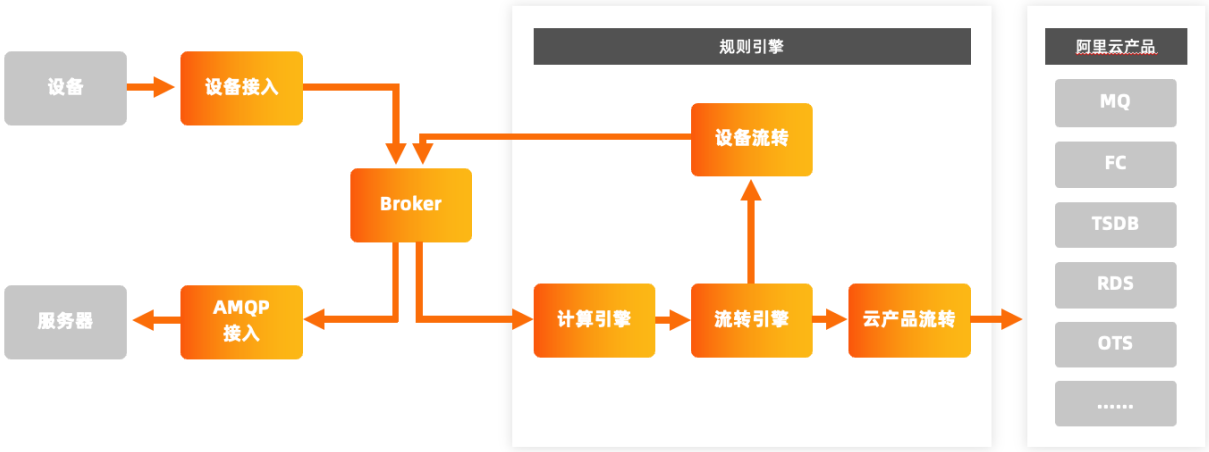


传统的MQ采用的拉模式，若出现消息堆积，后续的消息实时性会受到严重影响。IoT的业务场景对实时性要求很高，因此采用推拉结合的模式。

消费者通过AMQP协议与AMQP队列网关建立连接，设备上报的消息到达Broker后，Broker直接通过AMQP队列网关将消息实时推送给消费者，只有推送失败了才会通过队列进行重试。如果消费者不在线，则消息会堆积在队列中，等待消费者重新上线后，AMQP队列网关再从队列中拉取堆积的消息。

在这种架构下，推模式和拉模式结合了起来，链路上做了隔离，互不影响，并且保障了实时优先。

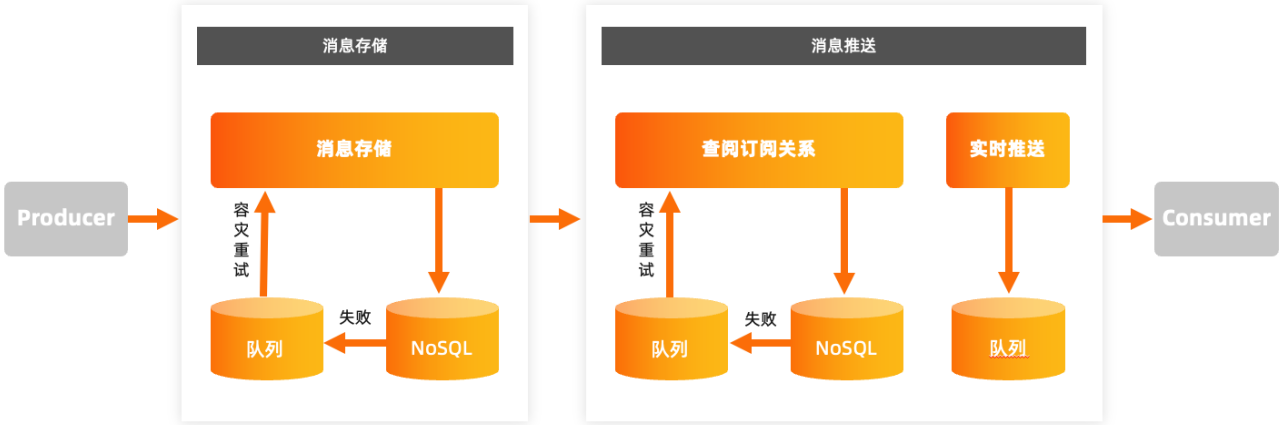
规则引擎



规则引擎提供了功能强大的SQL计算能力、消息过滤和加工的能力。通过规则引擎可以将设备上报的消息流转到各类阿里云云产品中，如消息队列、数据库、函数计算等。

规则引擎内嵌了自研的SQL执行引擎，支持通过SQL语法处理JSON、二进制格式数据。SQL函数提供了数学运算、字符串操作、日期操作等数据操作。具体在规则匹配时，计算引擎会进行词法分析、语法分析，解析出Action后交由规则引擎执行。

高可用



通过容灾、冗余、重试、隔离等多种技术手段保障高可用。支持百万消息并发，亿级消息堆积。

1) 容灾

IoT消息队列拥有一套完备的容灾能力，在面对亿级的流量洪峰、亿级消息堆积时，保障消息推送成功。

目前的容灾能力主要包括消息存储容灾、订阅关系容灾、消息推送容灾。

- 消息存储容灾：在存储引擎出现故障时，消息存储会不断重试，确保消息最终存储成功。
- 订阅关系容灾：消息推送的第一步就是查询订阅关系，如果存储引擎出现故障，Broker内部会发起重试，确保推送流程最终能正常走下去。
- 消息推送容灾：消息推送给消费者时，如果消费者出现异常，云端同样会发起重试，确保最终的消息推送成功。

2) 用户隔离

用户隔离是IoT消息流转高可用建设核心的架构优化。针对不同用户的SQL脚本、规则的复杂度等特性，用户隔离引擎能动态感知各用户规则计算的资源消耗，从而动态调整计算资源，保障负载均衡。

用户隔离引擎主要包括调度中心和状态中心。调度中心实时监听服务节点的状态，做流量统计，然后根据动态的一致性Hash算法做资源分配。状态中心则实时收集每个服务节点的状态，提供给调度中心做决策。最终调度中心决策出最优IP，指定IP调用。

3.2.3. 核心技术点

技术	说明
消息的可靠性	通过容灾、冗余、重试、隔离等多种技术手段保障消息的到达率。
千万级并发消息	架构水平扩容，支持千万级并发消息，万亿级消息堆积。
计算存储分离	消息存储与消息推送分离，吞吐量与性能大幅度提升。
亿级Topic	计算存储分离的架构，消息队列可以支持亿级Topic。
推拉结合	支持消息实时推送，支持消息离线拉取，推拉结合，实时优先。
计算能力	强大的SQL计算能力、自定义脚本能力、消息过滤与增强能力。

3.3. 物模型

3.3.1. 核心技术挑战

- 物模型的普适性

随着数字化的普及，越来越多的企业意识到设备数据上云的重要性，不同行业的企业客户，需要把海量的设备接入上云，借助物联网平台能力提升企业运行效率。不同行业场景设备复杂度、功能都不一样，从简单的智能家居设备（如智能灯泡），到工厂产线的复杂单体设备（如纺纱机），再到多种设备组成的复杂业务系统（如污水处理厂），设计一套足够描述海量设备的方法是物模型建设面对的首要挑战。

- 规则执行效率和稳定性

针对业务场景完成复杂业务系统模型构建，每个数字孪生体支持万级节点，每个节点支持300个属性定义，每个物模型属性又可配置10条数据计算规则。物理设备数据映射到孪生节点之后，自动触发多级节点间的数据运算。如何保障孪生体数据规则执行的效率和稳定性，为我们带来了很高的技术挑战。

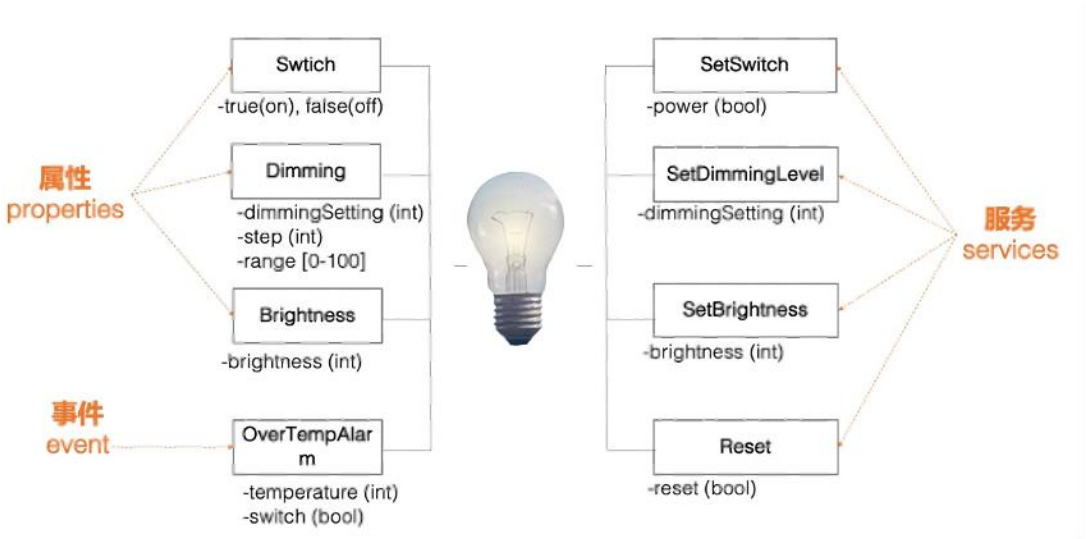
3.3.2. 技术详细描述

- 物模型的普适性

普适性要求物模型的能力要能覆盖工业、生活、农业、交通等各行各业多种不同设备，这要求物模型支持设备最本质的共性，抽象出一套模型，而且具备足够扩展性，可支持复杂的设备和场景的能力。

1) 首先想到面向对象的设计思路和开发语言。类比面向对象Java语言，用属性和服务来描述物的状态和行为，同时结合设备应用场景特性，抽象事件的概念。事件是一类需要客户及时响应的特殊属性，例如空调的故障告警，这类属性实时性强，一般需要监控并及时响应。

   以一台智能灯为例进行说明，其具备开关、色调、亮度、过温告警、恢复出厂设置等功能，其中包含有传感器采集的状态、有危险告警、和APP下发的控制指令。使用物模型属性、事件、服务能轻松描述该设备具备的能力。

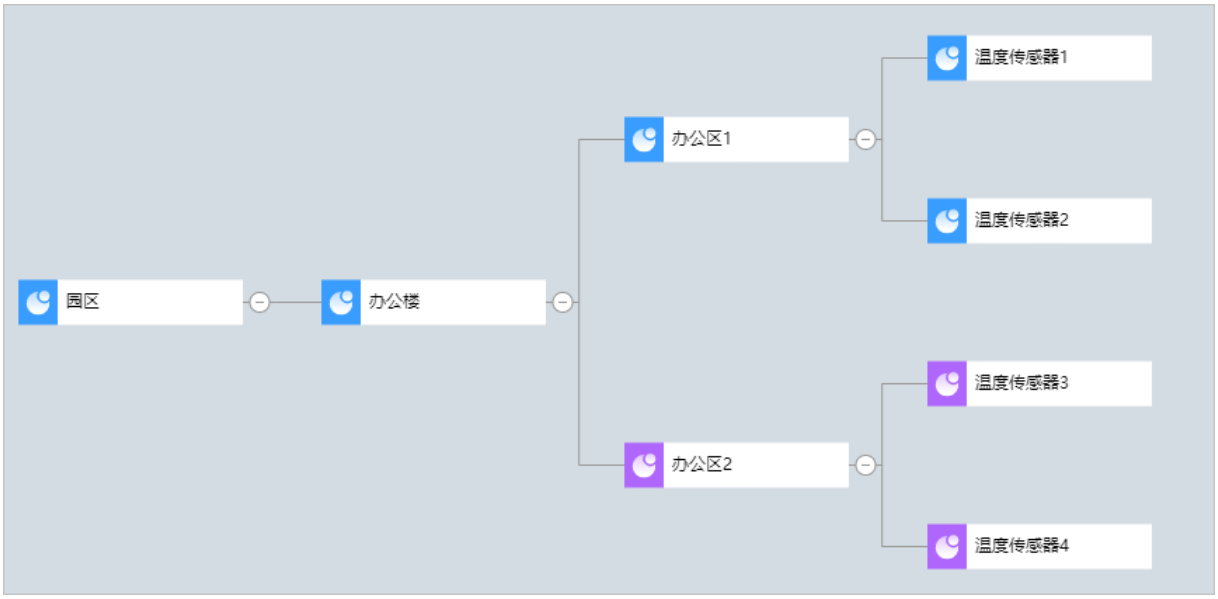


2) 针对每种数据类型还定义了非常严谨的数据规范，还需要定义数据范围、单位、步长等规范，例如当前智能灯的温度值取值范围为1至100摄氏度。

3) 联合芯片、传感器、模组、智能硬件等350多家IoT产业链合作伙伴，共同成立ICA事实标准联盟，沉淀标准物模型品类1000多个。

随物联网业务需求的发展，物模型的描述能力从单设备功能，逐步演进到描述某业务场景下不同类型多台设备组成的复杂系统，抽象数字孪生体的概念，描述多级设备间的树形关系结构，并配置运算规则，支持节点间数据运算。

以某园区温度统计进行简单说明：



（使用图数据库构建的园区设备关系，通过温度传感器计算办公区平均温度。）

1) 用户通过控制台可拖拽的方式构建数字孪生体，每个孪生体节点代表业务场景中的一台设备，每个节点可定制物模型功能定义和规则，物理设备采集数据可映射至孪生体不同节点，触发节点数据的规则运算。

2) 构建复杂业务场景多层级设备关系，底层存储从关系型数据库升级至图数据库，从树形关系结构升级至有向无环图，当前支持如体育场馆万级设备节点关系构建。

3) 孪生节点的多维度数据检索，包括节点ID、节点名称、节点路径、节点的动态属性值等。

4) 孪生节点间关系检索，借助图数据库边节点属性设置，支持祖先节点，子孙节点和兄弟节点检索和统计。

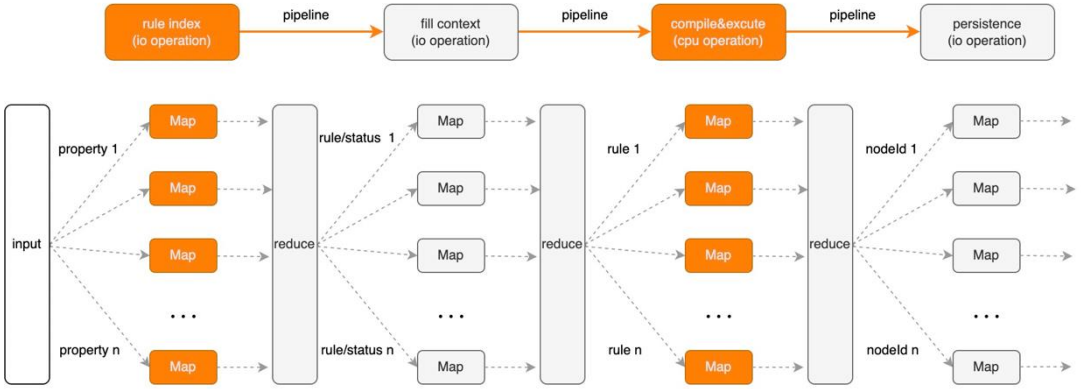
5) 可筛选符合业务规则的孪生节点进行批量控制。

- **孪生体节点规则执行效率和稳定性**

1) 通过异步数据流转对整个链路进行解耦，降低物理设备数据上报和孪生规则运算相互影响，提高链路的稳定性，借助规则引擎现有功能把采集数据映射到孪生节点，触发孪生节点规则运算，将运算结果数据写入到父节点，再通过异步消息触发父节点数据规则运算。通过异步消息对链路进行解耦，并借助运行态管控开关，调配系统负载和规则执行效率。

2) 避免规则环路检查。在管控态创建孪生规则时进行严格检查，避免出现规则环路，降低系统运行时风险。

3) 运行规则执行效率。通过规则索引表和多级缓存提升规则查询效率，并引入Aviator轻量规则引擎，对配置规则提前进行检查，并把运算规则表达式翻译成Java字节码执行，提高单条规则的运算效率。借助MapReduce数据集的并行运算，提升孪生体多条并发规则的执行效率。



4) 增加单独的监控预警能力，同时增加全局、用户、孪生实例多维度限流和降级策略，保障整体链路运行时的稳定性。

3.3.3. 技术核心点

技术	说明
基础数据能力	支持10种数据类型；支持2层复合数据类型嵌套。
复杂场景建模能力	拓扑关系：有向图；复杂度：10层；孪生节点：10000个。
数据计算能力	内置函数：25个操作符32函数；支持规则编译检查；支持脚本计算。
物模型描述语言	TSL描述语言。
物模型生态	主导ICA标准联盟，沉淀1000+标准品类，提供AIoT设备中心，认证硬件即插即用。

3.4. 大规模设备管理

3.4.1. 核心技术挑战

- **高效灵活的设备检索**

从设备管理运维的视角出发，除一般检索产品应具备的低RT、高QPS、高稳定性外，物联网平台还需为用户提供全面的数据、尽可能短的可见时延、灵活的查询，所面临的主要技术挑战包括亿级数据、数据高频变更、数据的时序特性、无冷热特征、结构松散和数据异构。

- **海量设备的管理运维**

物联网场景下，连接上云的设备数越来越多，当设备规模达到一定量级，如何对大规模设备进行管理和控制，例如大规模设备控制和远程升级，是一个非常大的挑战。这涉及到设备圈选能力、设备任务调度能力、云端向设备批量推送能力，并对系统稳定性也提出了更高的要求。

- **设备全球分发**

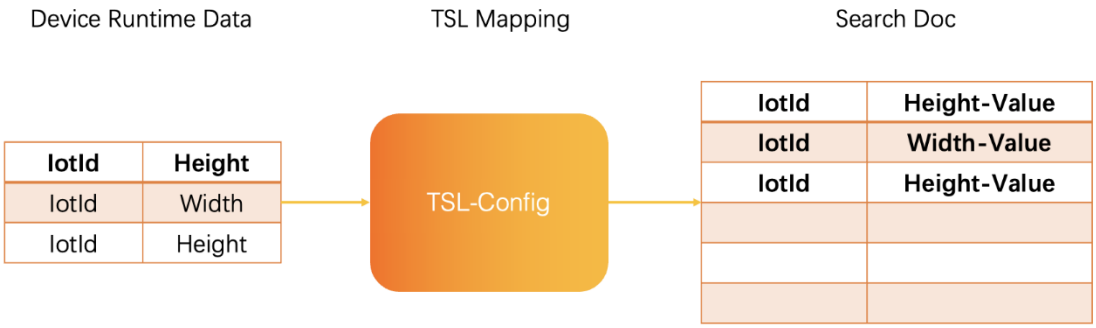
设备出厂常面临两个问题，一是在设备出厂时，需对设备的不同地域和不同实例的连接信息进行硬编码，导致厂商无法进行提前备货；另一个是设备生产者和使用者通常是不同的，需要解决设备的最终归属问题。所以我们面临的主要技术挑战是如何让设备不需要硬编码接入点信息就可以快速可靠地实现就近接入，以及如何实现跨租户、地域、实例复杂网络场景的大规模设备可靠分发的问题。

3.4.2. 技术详细描述

- **设备动静态检索**

IoT设备的大量静态元数据、运行时时序数据，组成了海量异构数据，为了从大量数据中快速高效检索出目标设备进行远程管理，物联网平台检索能力同时提供了动静态两种能力。静态检索，是基于已有数据检索符合条件的设备，动态检索则，是利用规则圈选一批符合条件（包括未来符合条件）的设备。主要包括的关键技术有：万级物模型索引配置、SQL-Like检索语法、动态分组圈选。

1) 万级物模型索引配置



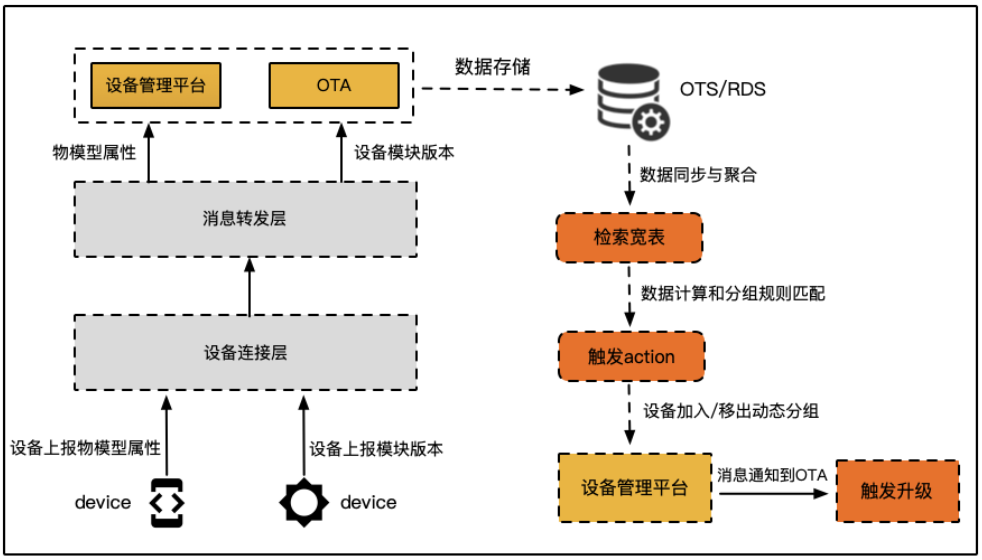
虽然单个设备的物模型属性数量是有限的，但是不同设备的物模型属性数是完全不一样的，这就导致最终设备的物模型的属性是不可穷尽的。但是索引表的宽度是有限的，因此就需要用有限的索引列存储无限的物模型数据。

主要的技术手段是通过结合物模型数据定义明确、整体数量不可穷尽、单设备可穷尽的特点，将单设备的物模型信息与索引进行映射，多设备复用相同索引，实现物模型数据的检索。

2) SQL-Like检索语法

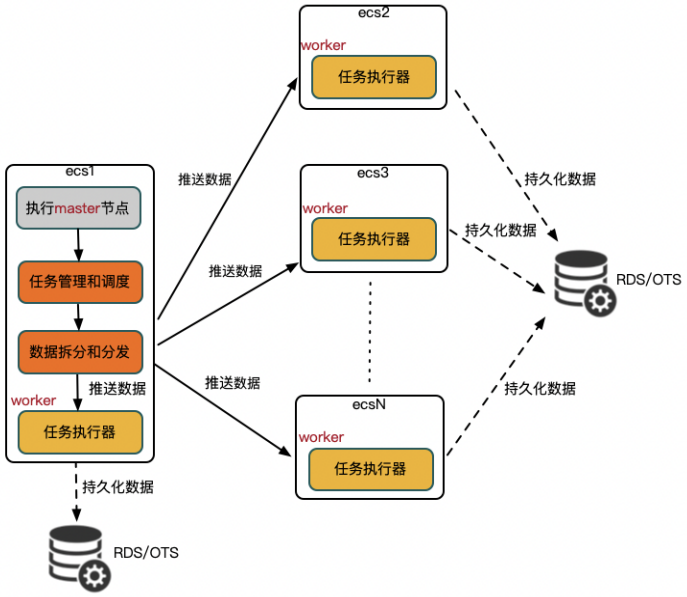
为了降低用户的使用成本，物联网平台提供了SQL-Like的检索能力，用户能够像查询数据库一样来检索数据。与此同时，底层使用了多套检索引擎，考虑到易用性在上层使用SQL检索的方式来屏蔽底层引擎的差异。物联网平台设计了一套可以适配多引擎和业务自定义的SQL检索框架。

3) 动态分组圈选



物联网平台支持基于产品、上传的DeviceName文件列表、灰度、地理位置、SQL检索、分组、标签、物模型属性、设备版本号等多种方式圈选设备发起OTA升级，提供丰富的圈选方式满足用户各种使用场景。除了使用到上述的SQL-Like检索能力外，还用到了基于动态分组的升级能力。例如建立一个分组是指定产品下电池电量大于80%的设备，OTA的升级策略是对该动态分组内的设备触发升级，其中电池电量可以是一个物模型属性，其技术难点是该分组下设备是随着物模型属性上报的实时值动态进行变化的，并非一个固定的设备分组。

- 大规模设备长任务的调度



大规模设备升级是一个典型的长任务，升级任务整体执行时间长，在升级过程中极易被中断。为了保证整个设备升级过程的高可靠性，需要使用长任务调度和管理机制。OTA不仅要支持单个批次下大批量设备同时升级，同时还需要支持多个大批量设备升级的并发升级。这对长任务框架提出了更高的要求：必须从传统的单机调度模式演进到集群调度模式，并在升级任务的调度、推送、执行上有更灵活的策略。

主要技术手段：

1) 动态调度

保证升级任务能够有序被调度执行，避免某个时刻发起升级的任务数过多导致系统出现瓶颈，当单租户下发起升级任务过多后会进入等待队列，后续达到运行条件后可被调度执行。

2) 异常自动恢复

动态检测运行中的任务，当发现任务被中断后可自动进行恢复。

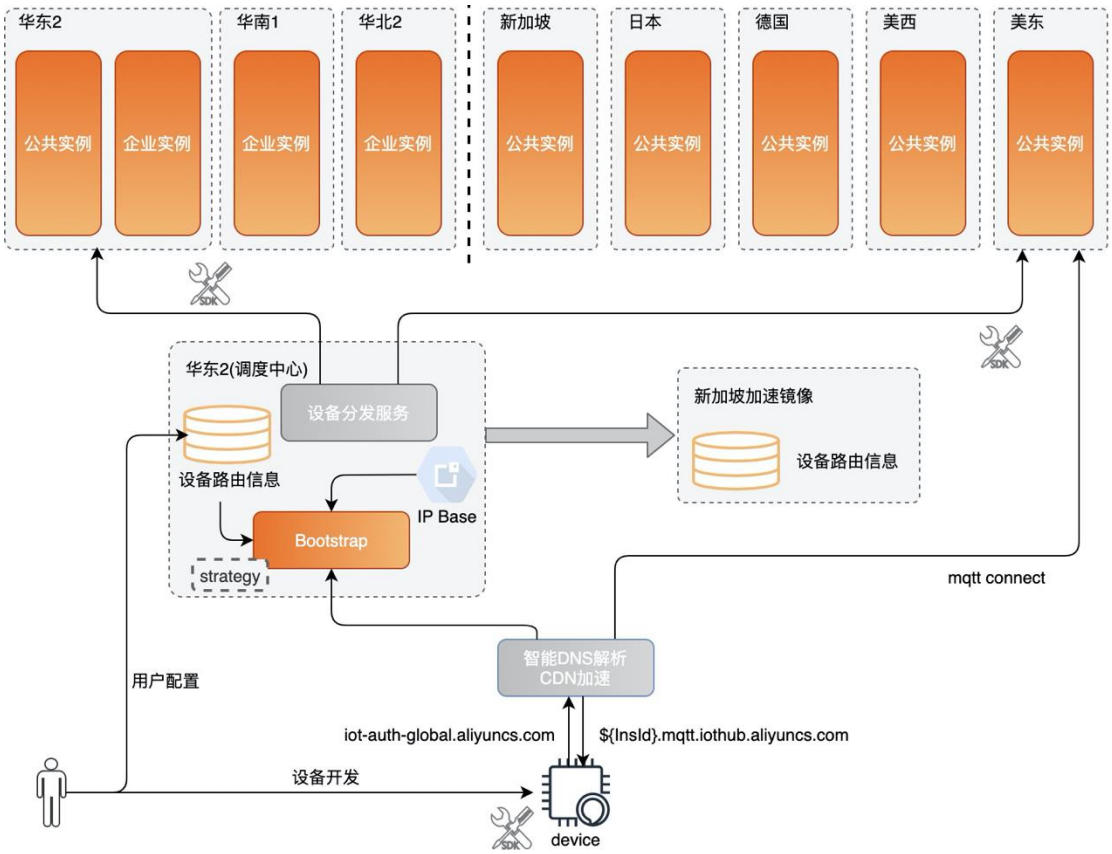
3) 分布式协同

充分利用集群能力处理大批量设备升级，其中设备圈选、升级任务初始化、推送升级消息三者可并发执行，从而可避免触发单机瓶颈，使大规模设备升级时系统各项水位更加平稳。

4) 精细化推送

在实现每分钟恒定推送速率的基础上，支持每分钟可变推送速率，可用于提高升级成功率。

设备全球分发



设备全球分发解决了设备出厂时，无需对设备的不同地域和不同实例的连接信息进行硬编码，仅需对设备统一烧录全球统一接入点信息（无地域信息）。设备出厂后，在物联网平台控制台对设备集中进行跨地域分发配置，实现设备全球就近接入。这里主要通过以下几个关键技术解决：

1) 网络互通

针对跨地域的场景通过跨域专线实现，针对跨VPC和经典网络访问通过反向VPC和AnyTunnel完成，实现了对用户屏蔽不同实例间网络的差异。

2) 分发任务管理

由于传输的数据量大且网络环境复杂，为了保证链路可靠性，可通过长任务异常恢复机制和事务来解决业务数据的最终一致性问题。

3) 分发策略

针对不同的业务场景分发服务提供了静态策略和就近接入策略，来解决设备跨租户归属和全球接入的问题。

4) 就近接入

设备只需烧录指定的全球接入点，请求Bootstrap服务，云端即可通过ADNS智能域名解析和CDN加速，将设备数据分发到就近地域并下发设备接入点域名。

3.4.3. 核心技术点

技术	说明
设备检索支持的字段数	100+。
设备检索的同步时延	10秒内。
设备检索性能	百毫秒级。
设备任务调度规模	10万。
任务推送速率	10000 QPM。

3.5. 监控运维

3.5.1. 核心技术挑战

- 灵活多变的监控需求

物联网平台监控场景面临的是上亿级别的海量设备，相比传统的IT运维，被监控的对象数量增加了好几个数量级。随着业务的飞速发展，面对平台动辄数十亿甚至百亿级别时序数据，我们该如何有效的监控与管理？而随着物联网时序数据量爆发式的增长，传统的线图、直方图、散点图等数据展示方法，很难直接让运维人员找到数据背后的异常或隐藏瓶颈。如何针对不同业务或者不同监控对象，找到更合适的数据看板以及展现形式，成为物联网平台必须解决的问题。

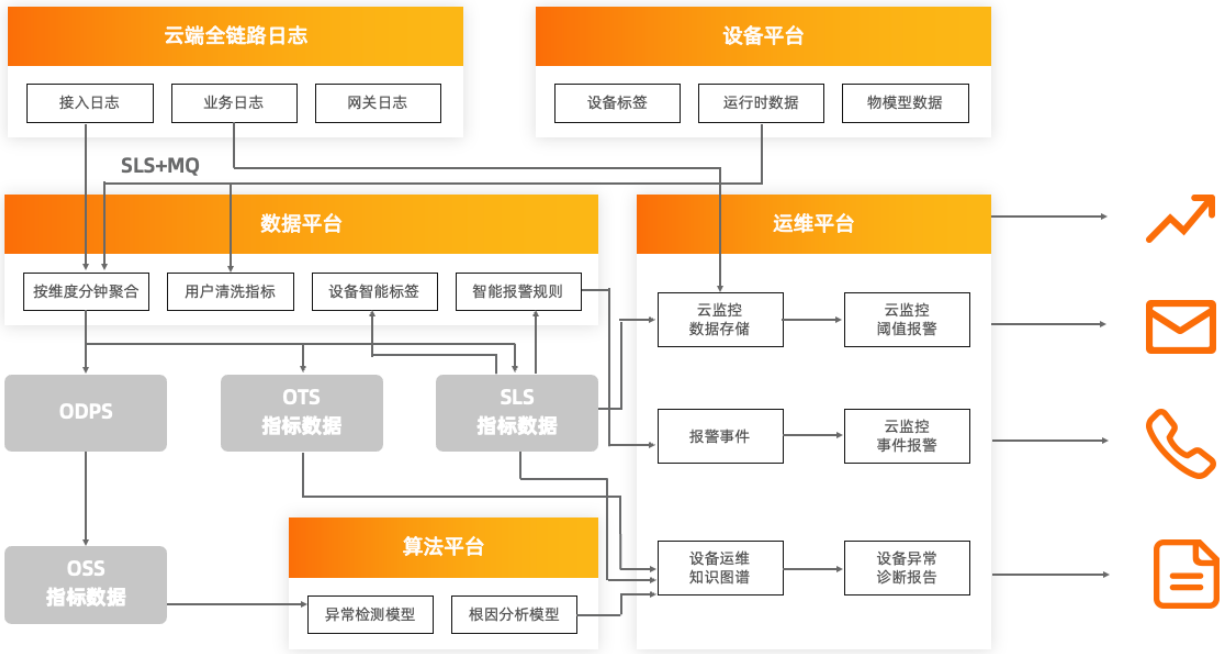
- PB级日志数据分析处理

物联网平台是一个复杂的分布式系统，设备消息上下行、设备控制链路都非常复杂，涉及到了非常多的云端系统。而传统的日志信息也往往有多个来源，例如营销活动打点、用户访问、应用日志，并且来自于ECS服务器、容器、移动端、网页端等多种渠道，需要多渠道、多维度、多种处理方法。

物联网场景中日志系统面临着极大的规模挑战，面对上亿在线的设备所产生的数据，系统应具备利用这些日志快速解决问题的能力，这也就要求系统能处理大量数据，且实时性要求高。同时，为了充分发掘日志内容的业务价值，需要结合设备运维场景对多渠道日志做全面分析，监控异常设备指标，定位系统问题，分析出相应的异常调用链路。

3.5.2. 技术详细描述

自定义监控大盘



传统的物联网平台提供了诸如实时在线设备、上下行消息总量、规则引擎消息流转次数等有限几个系统指标，只能满足客户的基本运维需求。客户根据不同的业务需求，需要监控的数据指标往往存在差异，传统的实时监控指标很难满足客户的日常运维需求。

物联网平台的自定义监控大盘提供了设备、消息、物模型、规则引擎和OTA升级相关指标数据的实时监控服务，指标维度可以选择物联网平台的所有产品或指定的单个产品，指标聚合支持最小、最大和平均等聚合方法，聚合粒度可选择不同的时间频率，基本满足了客户日常运维的刚性需求。

为保证客户最佳的实时监控使用体验，这里主要通过以下几个关键技术，解决实时监控所面临的数据规模和个性化所带来的技术挑战：

1) 链路的规范性

数据平台对ODPS离线数据和SLS实时数据进行实时筛选、合并和计算，统一计算后将结果输出给云监控，从而实现了系统指标和用户自定义指标数据链路和指标计算的统一性。

2) 计算的实时性

数据平台引入实时数仓Hologres，对衍生指标和原始指标建立了全局加速表，实现了数据的全链路实时化改造，通过Flink将指标聚合计算做到了秒级延时，将原来报表的展示延时从30秒下降到1秒以内。

3) 诊断的联动性

针对大盘指标提供了可配置可组合的下钻能力，帮助精准圈选出故障相关的异常数据，让数据下钻和后续算法平台输出的根因分析模型形成了联动，既可以帮助发现数据共性，同时还能缩小后续故障分析中的数据计算量，一定程度上提高计算效率。

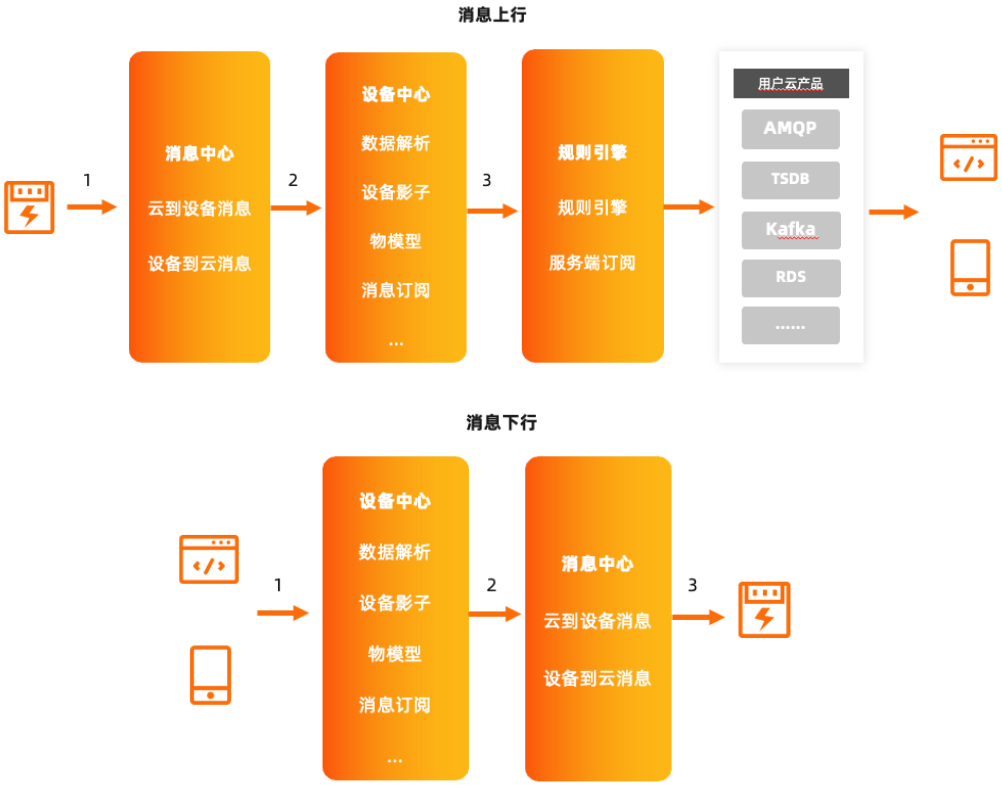
消息轨迹

消息轨迹解决了上下行链路中问题定位的难题，客户可根据TraceID或MessageId，追踪任意一条消息在物联网平台流转的全路径，还可根据出现的故障节点快速分析、定位问题。为保证最佳的链路诊断使用体验，这里主要通过以下几个关键技术，解决消息轨迹所面临的日志存储成本、时序错位和查询性能瓶颈所带来的技术挑战：

1) 链路业务抽象

物联网平台是一个复杂的分布式系统，设备数据上报、下行控制链路都非常复杂，涉及到了云端非常多的内部系统，这些复杂度无需暴露给用户，用户无法理解，对问题分析也会存在干扰。另外，系统内部的日志输出内容较多占用了非常多的存储，格式本身也不统一，随时可能存在变化，从而无法针对日志做深入的分析。

为消除用户的理解成本，在分布式链路中，物联网平台梳理出了消息上下行链路中的关键系统，按照日志聚合规范输出了关键调用节点信息，面向客户可理解的业务原语输出了诊断信息，帮助客户快速识别出链路上的异常节点，并根据错误码的提示进行问题诊断和修复。



2) 推断链路时序

全链路租户日志信息来自于多个不同的分布式系统，打印日志的时间戳非常接近，消息轨迹采集的是按规范输出的业务日志，将系统、模块之间的调用逻辑顺序通过规则配置沉淀下来，通过TraceId或MessageId获取具体租户业务日志时，可以根据逻辑顺序重新绘制出调用链路时序，而无需依赖时间戳。

3) 提升查询性能

租户日志虽然是针对业务特性精简过的日志，但物联网平台上下行链路每天都会产生巨大的日志量（PB级别），考虑到日志存储成本当前物联网平台只会存储7天的租户日志，即使这样，在所有的租户日志中查询一条特定链路的信息也面临精准性和性能的挑战。针对租户日志查询条件，平台对字段进行了索引加速查询性能，同时也支持用户将租户日志导出到自己的SLS空间长期保存。

3.5.3. 核心技术点

技术	说明
自定义监控大盘	提供监控指标近百个，支持包括总和、最大值、最小值、平均值4种常用的聚合类型，指标聚合计算可做到秒级内延时，支持基于物模型自定义指标，对指定设备进行精细化运维管理。
消息轨迹	提供云端全链路日志查看设备通信消息轨迹功能，可根据TraceId或MessageId追踪任意一条消息在物联网平台流转的全路径，还可根据出现的故障节点快速分析、定位问题。
故障诊断	离线分析设备大批掉线原因、离线设备区域分布以及掉线未重连列表，为运维人员后续处理提供决策依据；诊断设备异常情况全景并给出诊断报告，分析高频的异常类型并予以排查方向建议。
远程隧道	Linux系统设备可直接集成SDK通过控制台远程SSH登录，对于非Linux系统（如Windows、RTOS、uCOS等）可基于远程隧道搭建设备远程访问能力，支持运维通道与数据通道的隔离，提升稳定性。

3.6. 异常检测

3.6.1. 核心技术挑战

- **海量设备的实时流式检测**

  物联网平台的接入设备数量极大，传统的手工运维方式无论是准确性还是人力投入都无法满足需求。近年出现的若干基于统计算法的异常检测模型，往往需要一个设备一个模型，设备数量较大时模型计算开销极大，并且此类统计算法缺乏实时性，难以处理流式数据。

- **为时序预测算法赋予时序检测能力**

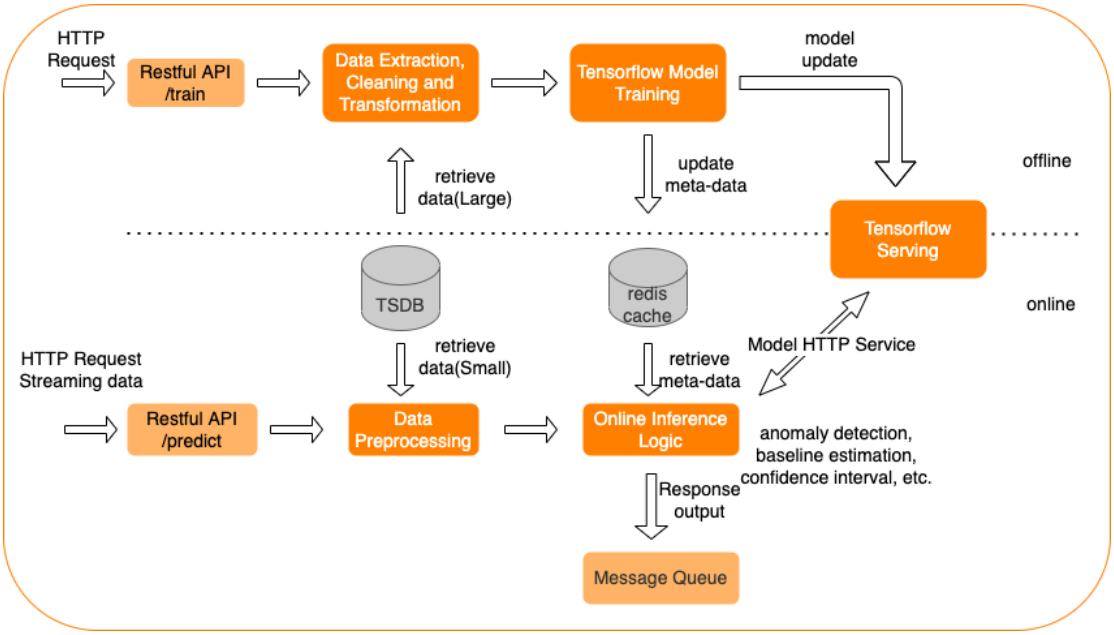
  基础算法DeepAR的提出是为了解决时序预测的问题，属于回归分析。但在时序异常检测中，最终的结果是Yes或No的异常判断，是分类问题。在异常检测中，历史数据的噪音也会对检测的结果造成影响。为了提高模型的0/1二分类能力，同时压制数据噪音带来的过拟合影响。

- **非平稳时序数据的建模**

  无论是统计时序算法还是神经网络时序算法，均要求时序数据的平稳性。然而在IoT场景下存在大量非平稳数据。

3.6.2. 技术详细描述

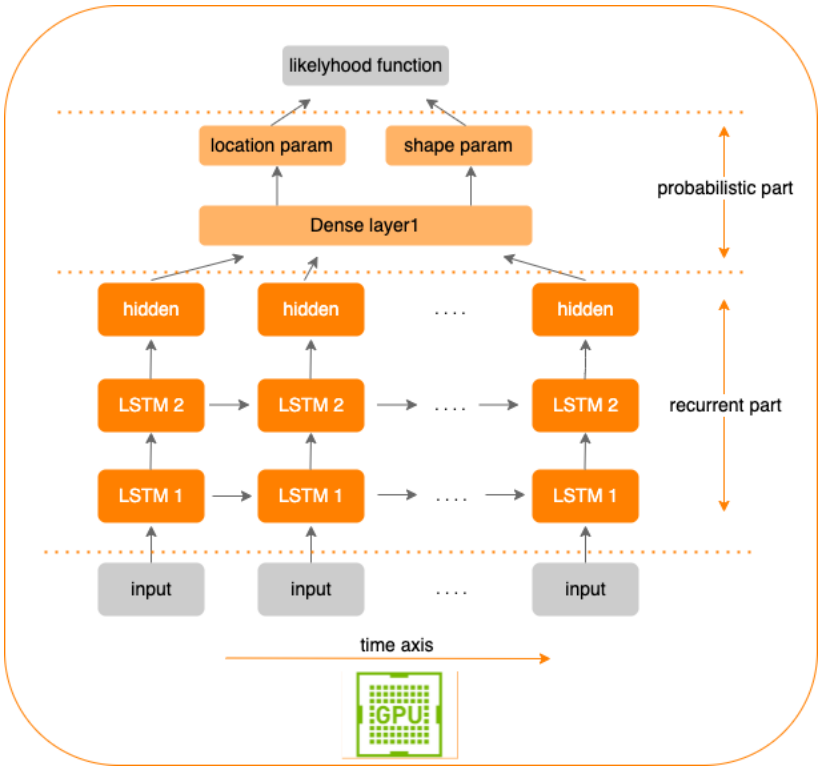
为了应对第一个挑战，需要设计一个能够处理海量设备，并针对IoT数据呈现流式时间序列的特点开发异常检测算法。基于此考虑设计的人工神经网络算法，通过提取历史数据的时序特征，动态地计算上下限阈值边界，将多设备、多指标统一到一个模型中，可以实时捕获异常并进行报警，提高设备运维效率和质量。



系统分为离线训练（offline）和在线推理（online）两部分。离线训练部分定期执行（如每日凌晨），从时序数据库中抽取用户近14天的历史时序数据，输入算法进行训练得到模型。在线部分每分钟实时调用，从时序数据库中拉取用户最近2小时的数据，经过标准化等操作后作为算法输入，算法计算得到当前的基线值和上下限置信区间。如果当前真实值突破了置信区间，则有较大的概率说明当前值不符合历史趋势，出现了跳变，从而触发用户告警，写入消息队列通知用户。

3.6.2.1. 算法原理

模型采用基于概率自回归的神经网络，基础框架来自于论文“DeepAR：Probabilistic Forecasting with Autoregressive Recurrent Networks“。



算法框架主要包含三个主要步骤：

1) 利用循环神经网络LSTM对历史时序数据进行特征提取。

2) 将提取的特征映射到高斯分布，并获得位置参数和形状参数。

3) 计算当前点的Gaussian log likelihood，采用极大似然法优化整个网络的参数。

在应对算法开发的第二个技术挑战中，我们在原始论文的gaussian log likelihood损失函数的基础上，引入了基于bernoulli likelyhood的loss function。

在实际中，与告警的敏感度相关，可以将其设为异常检测的敏感度水平，如1%、3%、5%等，分别对应低、中、高敏感度。

将b_log_likelihood加入基于高斯的g_log_likelihood，让模型对二者进行同时优化，可以将模型的二分类稳定性大大提高，并且满足我们预设的敏感度水平。

此外，应对IoT非平稳数据的第三个挑战中，我们对原始数据进行一阶差分操作，让神经网络对一阶差分后的平稳数据进行拟合。

$$z_i = y_i - y_(i-1)$$

对差分进行拟合后再进行积分操作（累加）进行还原便可以得到原始的预测。

另外，为了进一步提高算法报警的精确性，降低误报率，我们设计了算法的后处理操作，对误报进行抑制。

在大量的IoT时序数据中我们发现，许多跳变的尖刺发生后，时序会短时间恢复。但是时序预测算法的惯性会导致恢复过程被检测为突变的异常。为此我们设计算法如果发现属于时序恢复现象，则对告警进行过滤，消除误报。

超参数的选取对模型的最终结果有很大的影响。为此我们设置了超参数空间，并基于hyperopt对超参数进行自动筛选。超参数空间包括：

1) 模型自身参数：如层数，神经元数。

2) 训练过程参数：批大小。

3) 防止过拟合参数：dropout, regularization等。

训练区间为过去14天到过去1天的历史数据，验证区间为最近1天，获得最优参数后再进行模型重训练。

3.6.3. 核心技术点

目前物联网平台统计报警的精确率和召回率分别在90%和75%左右。但是需要指出的是，异常判断存在一定程度的主观性，目前业界尚没有一个完全通用的判断标准。为此，我们在服务的设计中引入了多种敏感度的设定，让用户能够控制、调整算法的策略，提高对最终结果的解释性。随着敏感度的提高，报警的召回率会不断提高，而精确率会出现下降，这也是符合逻辑的。

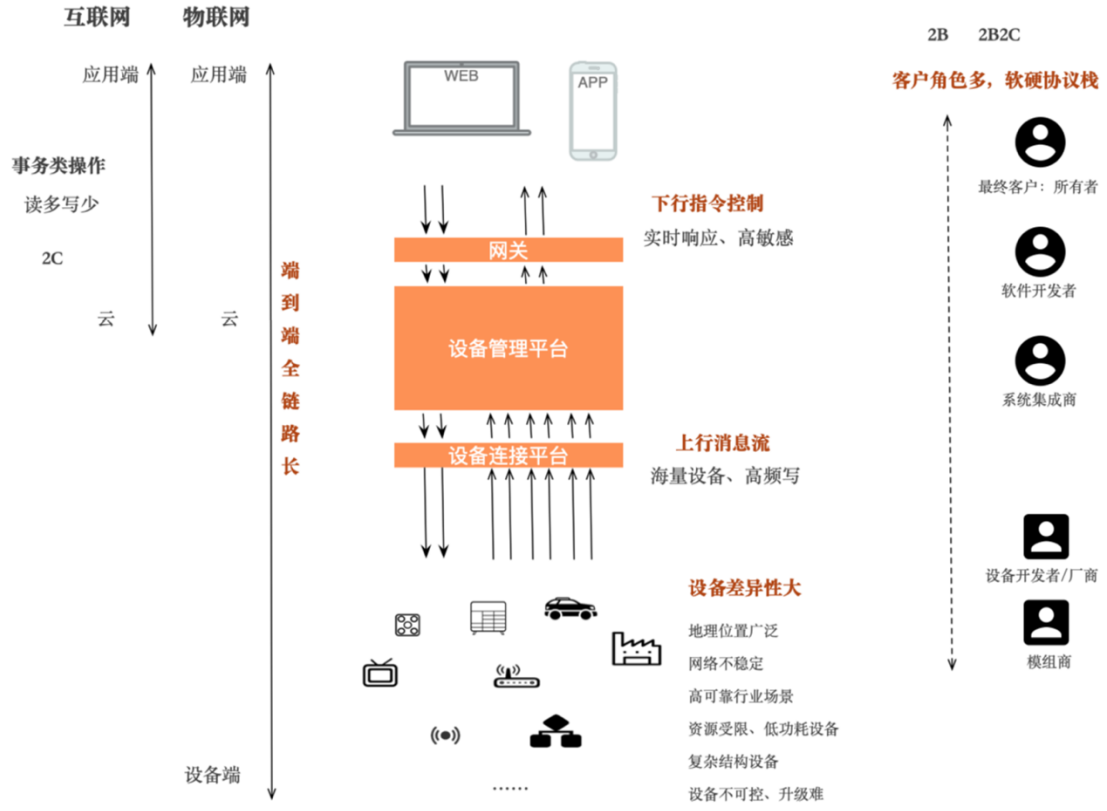
技术指标	告警级别：低	告警级别：中	告警级别：高
Precision精确率	0.978	0.905	0.79
Recall召回率	0.71	0.77	0.82

04 高可用

4.1. 重要性

设备连接、管理和运维作为PaaS能力赋能物联网客户，设备所属行业场景多，会被用于城市交通、工业制造、企业办公、商场电梯等场景，若出故障，可能会导致工厂停产，公交瘫痪，电梯停运等情况，对社会秩序和民生安全有非常大的影响，稳定性要求要远高于一般的互联网场景。

4.2. 挑战性



1) 端到端全链路长，移动互联网在互联网的基础上扩展了APP端，物联网又扩展了设备端，APP端、云端、设备端三端组成的链路非常长。例如一次下行控制指令，需要从APP到设备的一个来回，对全链路高可靠、可运维都带来了很大的挑战。

2) 设备弱网、资源受限、端侧固件异常、运营商网络抖动等情况，都可能对这种大规模流量链路带来雪崩效应，同时物联网设备下行控制指令对实时性和到达率要求极高。

3) 设备上行消息典型的高频写、写多读少、时序性强，与互联网应用流量读多写少完全不一样。

4) 设备因碎片化和场景化特性本身差异性极大，不像移动互联网手机相对标准，例如地理位置、网络、资源、结构、固件等都会对稳定性产生影响。

5) 物联网稳定性建设面对的角色也比较多样，很多情况需要应对不同稳定性要求的角色提供不同的高可用能力，并且要充分考虑设备程序质量从而规避云端风暴。

4.3. 单元化结果

设备连接和管理服务除了在每个模块进行独立的稳定性建设，例如连接的发布不断连、消息的容灾和重试、数据存储的异构容灾等之外，也升级了单元化的高可用架构，可通过单元化隔离和容灾，降低故障爆炸半径和应急快反速度，同时支持灰度、弹性、数据安全、蓝绿发布等能力。

4.3.1. 技术挑战

1) toB业务

   toB业务的流量分片很容易带来大租户、流量不均衡等问题，大租户问题会进一步引发分片容量规划和水平扩展问题。

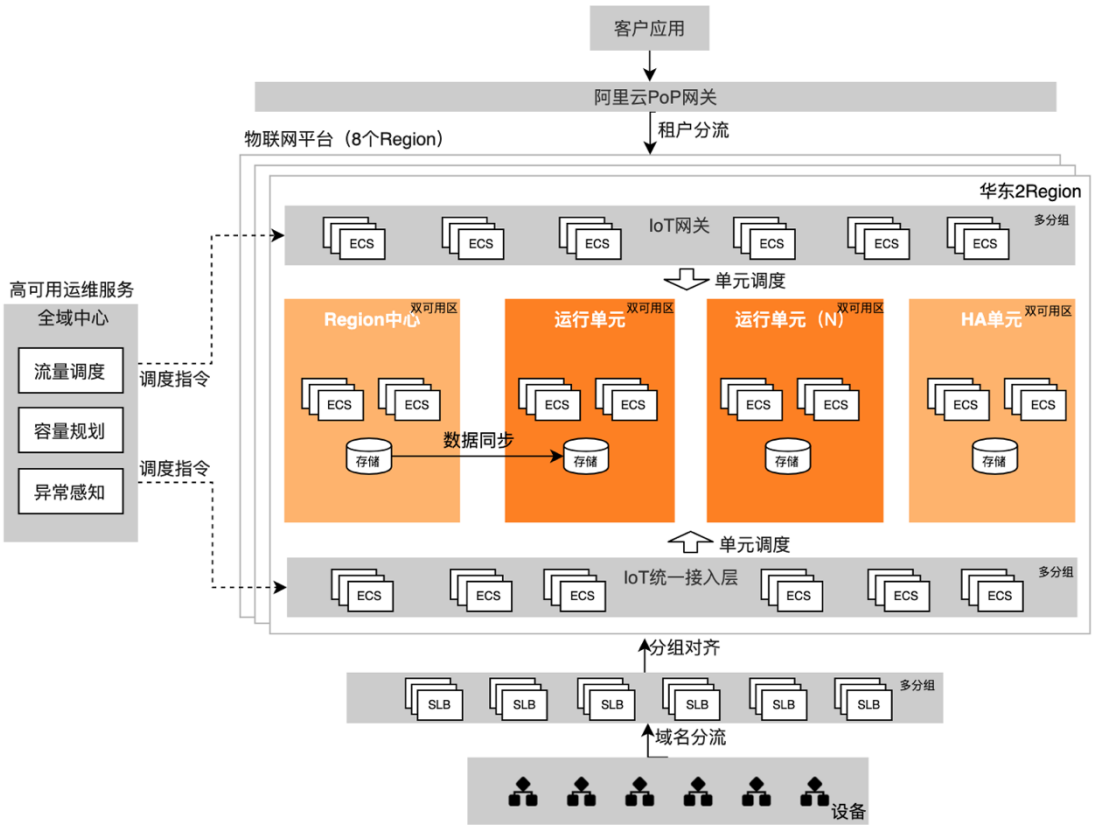
2) 物联网设备

   不同于互联网业务，物联网除了应用（Web/APP）、服务端外、还有设备端，北向应用和南向设备流量调度需要保持一致，单元化封闭需要考虑端到端全链路，同时容灾切流也需要保障端到端全链路流量一致性。

3) 阿里云Region化架构

   阿里云天然Region化架构，本质上是一种特殊的单元化，不过Region粒度太粗，因此在Region化架构下，我们进一步支持了更细粒度的业务单元化，既遵循阿里云多Region模式，又扩展了业务单元做好弹性和隔离。

4.3.2. 核心思路



在阿里云原有Region化架构下，将单Region流量进一步按照租户或实例打散。原来单Region架构拆分为运行单元、HA单元、Region中心三个核心单元。运行单元运行的是租户或实例内完全封闭的核心功能，Region中心承载部分无法分散到单元内的功能，例如跨租户功能、管控能力，HA单元作为运行单元的容灾能力，在运行单元出现异常可以快速FO。

4.3.3. 核心技术点

技术	说明
百万长连接分钟级切换	物联网平台统一接入层实现了百万长连接设备分钟级切换能力，采用的是智能路由和Session迁移能力的组合，同时根据设备量和切换时间实时计算切流速度。
API网关秒级路由	API网关实现北向流量按照租户维度秒级切换，对短连接实现真正的秒级生效。
流量调度	通过高可用运维服务实现故障快速检测、故障感知以及单元流量切换，流量秒级切换能力已经具备。
单元容灾	通过部署HA单元，实现与运行单元双副本互为FO，运行时支持数据同步，并做到严格的数据一致性保障，同时HA过程可以做到平滑切换，客户应用和设备无感知。
数据同步	通过流量标记方案，在业务侧无感情况下做到低延迟数据双向同步，无循环风暴，同时使用准实时的双向缓存预热和清理机制，保证容灾单元缓存常热，各单元缓存数据一致。

阿里云物联网平台：更快，更稳，更安全

主编：熊益群、王鹏飞、朱江、张宇

编委：李彬、张程、刘思谦、陈海滨、王霏、王明、葛成、王麦棋

监制：王进、何云飞、常司晨

设计统筹：仲祐民

市场推广：林紫玉、张婉莹、吴娴斐

出品团队：阿里云智能IoT事业部