

物联网平台

IoT Platform

技术白皮书



1 行业趋势

- 1.1. 物联网发展趋势 02
- 1.2. 面临的技术挑战 02

2 技术架构

- 2.1. 背景信息 03
- 2.2. 产品介绍 04

3 技术优势

- 3.1. 安全稳定连接 05
- 3.2. 海量消息处理 09
- 3.3. 大规模设备管理 13
- 3.4. 设备建模 16
- 3.5. 监控运维 20
- 3.6. 数据服务 23
- 3.7. 数据智能 27

4 全球化

- 4.1. 重要性 34
- 4.2. 挑战性 34
- 4.3. 核心思路 34
- 4.4. 核心技术点 36

5 高可用

- 5.1. 重要性 37
- 5.2. 挑战性 37
- 5.3. 高可用演进路径 38
- 5.4. 单元化架构 39
- 5.5. 稳定性体系建设 41



01 行业趋势

1.1. 物联网发展趋势

物联网已经经历了超过十年的快速发展，尤其是最近几年，物联网各个领域需求旺盛，各式应用场景愈加丰富，技术和应用创新层出不穷，发展速度越来越快。预计到2025年，全球物联网总连接数将达到246亿，年复合增长率高达13%，并且产业物联网设备数量将超过消费物联网设备数量。

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。物联网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. 面临的技术挑战

随着大量物联网场景不断涌现，海量设备的管理运维、海量数据的分析处理，以及业务全球化的发展趋势，都给物联网平台带来了一系列新的要求和新的技术挑战。

➤ 高并发

物联网设备规模随着行业发展不断增多，设备逐步增多要求平台的系统容量能不停的水平扩展，需要使用什么样的系统架构来支撑无限扩展的系统？对于国内最大的物联网平台来说，怎么解决千万级设备的并发建连、怎么维持百亿级设备的长连接稳定性、怎么保障百万级/千万级消息通信的实时性是个非常复杂的系统工程。

➤ 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，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。

➤ 数智化

业务数字化转型最终的目标是为了智能化，利用大量数据分析进行经营提效、降低成本、创新业务，物联网也开始逐步进入到数智化的阶段。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。

➤ 全球化

近年来，随着国内工业制造水平的提升和低成本产业链的优势，越来越多的企业将业务拓展到了海外，如汽车、光伏、储能、智能家居等场景，对物联网设备出海的需求日益增强。百万级设备在全球范围内的接入、管理和运维将给企业带来巨大的技术挑战，包括性能、稳定性、安全合规、运维成本等诸多方面，如何以极致的成本保障业务的安全稳定运行是企业业务全球化发展过程中首要考虑的问题。

02 技术架构

2.1. 背景信息



物联网产业由“云”、“网”、“边”、“端”、“用”五大板块组成。

“云”：指物联网相关的云化能力平台，主要由云端物联网平台承载相关云服务包括连接、管理、数据、AI、安全、区块链等等。

“网”：连接网络及相关产品和服务。大物联时代带来的大连接数和复杂设备现场环境，使得有线连接捉襟见肘，因此在AIoT应用场景中，网络将逐步以无线连接为主。

“边”：泛指中心节点之外的位置。边缘计算指的是将计算及相关能力从中心处理节点下放至边缘节点后形成的，靠近终端的计算能力。

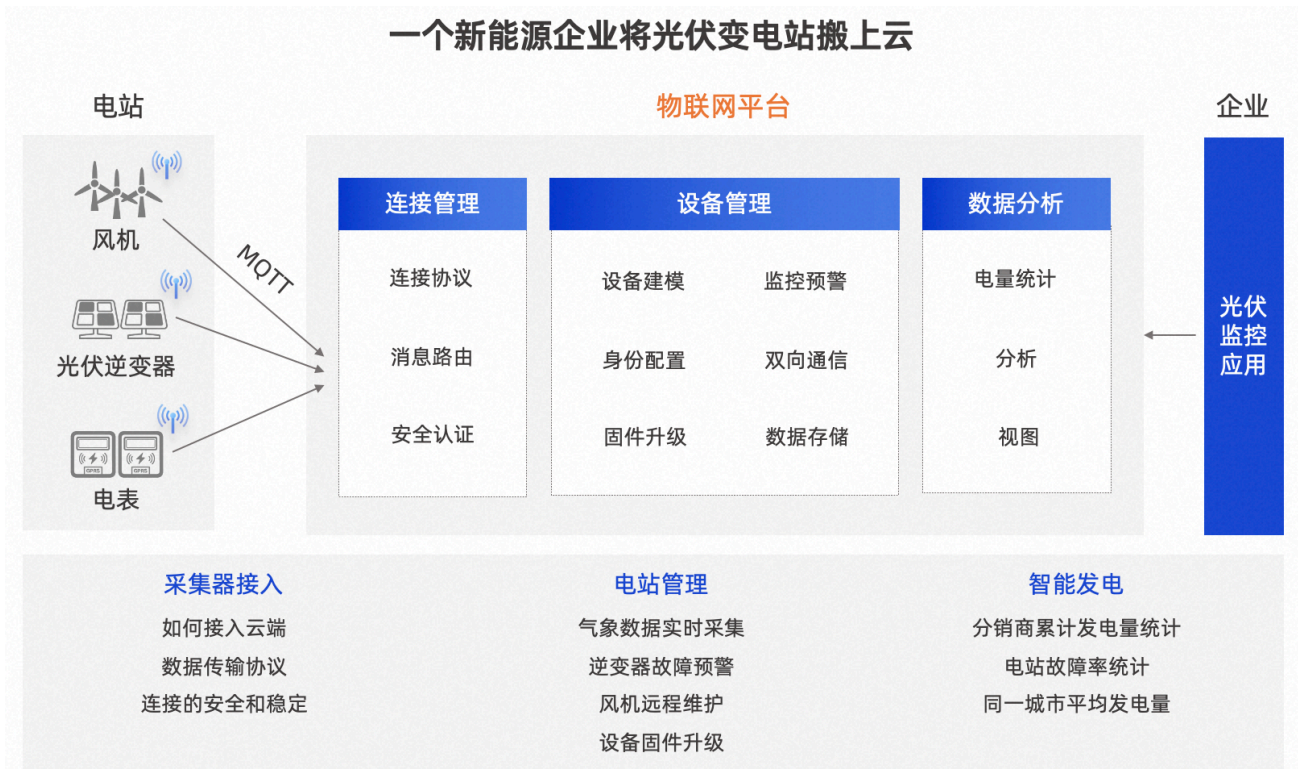
“端”：物联网终端，包括底层的芯片、模组、存储设备、感知设备、AI芯片、操作系统等。

“用”：AIoT产业应用行业。从核心驱动要素看，可分为消费驱动型、政府驱动型和产业驱动型行业。

2.2. 产品介绍

阿里云物联网平台提供全托管的企业级实例服务，具有低成本、高可靠、高性能、高安全的优势，无需自建物联网基础设施即可接入各种主流协议设备，管理运维亿级规模设备，存储备份和处理分析EB量级的设备数据，帮助企业快速实现设备数据和应用数据的融合，实现设备智能化升级。

以光伏逆变器为例，如果客户要自己实现设备在线化、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、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。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。



➤ 设备连接通信

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、订阅灵活性的要求与互联网面向人或应用的消息特点存在明显差异。

➤ 设备管理运维

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。设备建模将设备映射到云上产生孪生体，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特性，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。

➤ 设备数据处理

设备的智能化，最核心技术在于数据解析、数据存储与分析洞察。数据解析将多种多样的物联网自定义业务格式的数据进行实时的格式解析和预处理，产生适合于存储和后续分析的规范化的数据。数据解析需要考虑实时性、稳定性，并支持尽可能多的协议格式。数据存储支持存储用户上传的物模型数据，也支持存储经过数据解析后的规范化数据。数据存储需要充分考虑用户多种使用场景，包括为用户业务系统提供低延时高并发的读写服务、进行海量数据分析、进行低成本归档存储等。要充分发挥存储在物联网平台数据的价值，需要系统能支持常用的数据分析手段、AI算子，能方便地搭建数据处理流水线。同时，系统计算资源需要有足够强的弹性，以满足海量数据计算的资源消耗。

以下会从设备连接通信、设备管理运维和设备数据处理中，选取7个关键技术进行解读，包括安全稳定连接、海量消息处理、大规模设备管理、物模型（设备建模）、监控运维、数据服务和数据智能，以及物联网平台高可用和全球化技术架构的相关内容介绍。

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

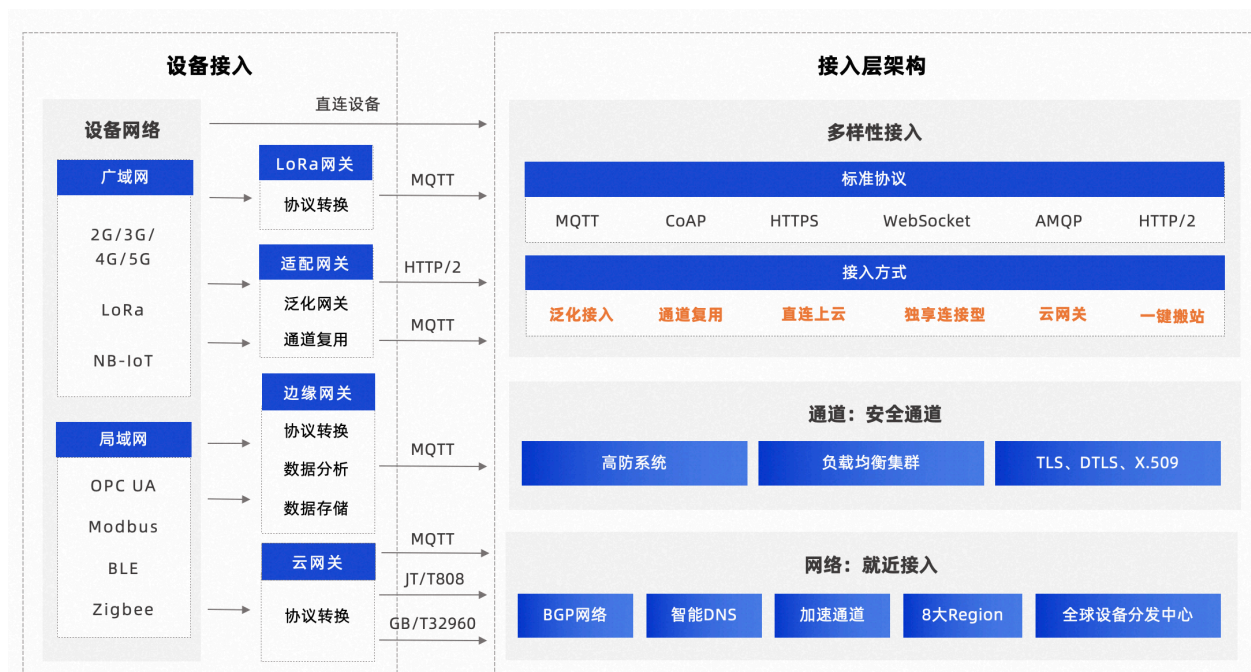
设备种类广泛导致设备部署的位置不同，例如在地下、在高空、在荒野、在边缘地区或在海外等，怎么让不同设备都能有好的网络质量，是接入层首要解决的问题。因为设备连接上云是IoT的基础。

➤ 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

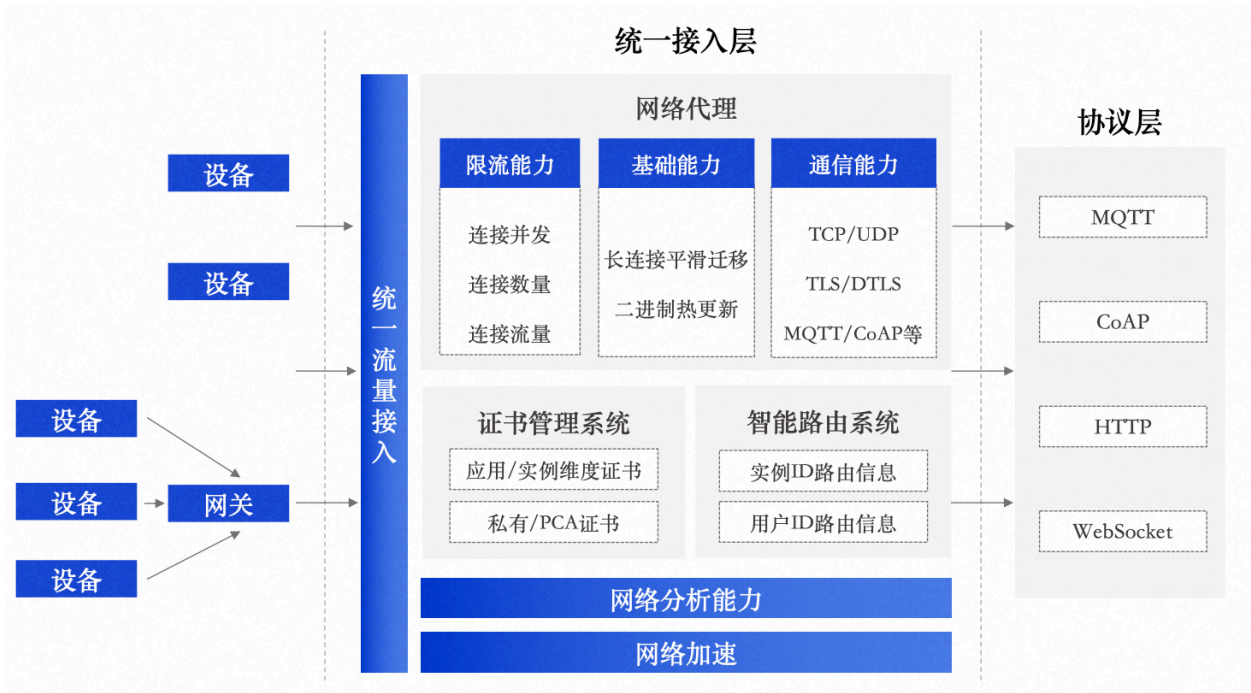


针对碎片化的接入问题，通过不同的接入方式来适配不同的设备场景。

- 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，由边缘网关转换协议，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：阿里云物联网平台支持移动、联通和电信NB设备接入。移动和联通可直接接入，电信支持两种接入方式。第一种通过Host/IP白名单接入阿里云物联网平台，第二种通过阿里云物联网平台提供的云云对接的方式，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复用网关的物理连接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。采用JT/T808、GB/T32960协议的车载设备，通过云网关解决认证、身份和设备通信问题。

➤ 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）和节流机制（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

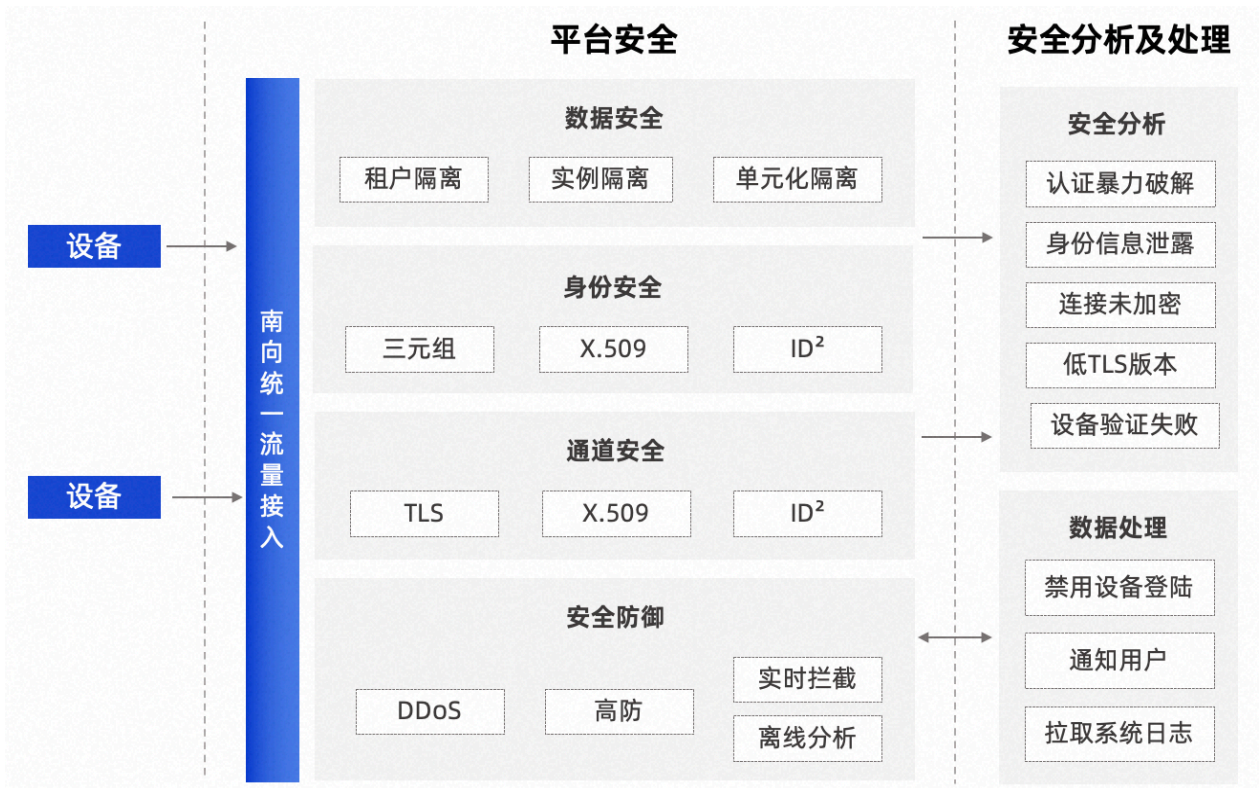
为了解决单故障导致的全平台问题，阿里云物联网平台针对协议层、消息层部署了多个集群，同时会把不同的用户放到不同的集群里，当某个集群的协议层或者消息层出现系统异常时，网络代理层可以通过路由能力和Session转移能力把设备的连接转移到不同的集群，从而保障单集群故障能快速恢复。



➤ 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，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，并且结合平台安全层，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. 海量消息处理

3.2.1. 核心技术挑战

➤ 高性能

物联网消息队列对性能要求非常高，业务场景需要支持千万在线设备，百万消息并发，毫秒级消息延迟，亿级消息堆积等。这些都是很大的技术挑战。

➤ 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，是IoT消息队列的核心挑战，也是最基础的能力。

➤ 秒级消息广播

秒级消息广播：广播能力是消息队列的一个重要能力。在物联网场景下，需要支持将消息广播给百万在线设备，且实时性要求非常高，如地震预警。如何做到秒级消息广播给百万在线设备，是IoT消息队列的核心技术挑战。

➤ 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。

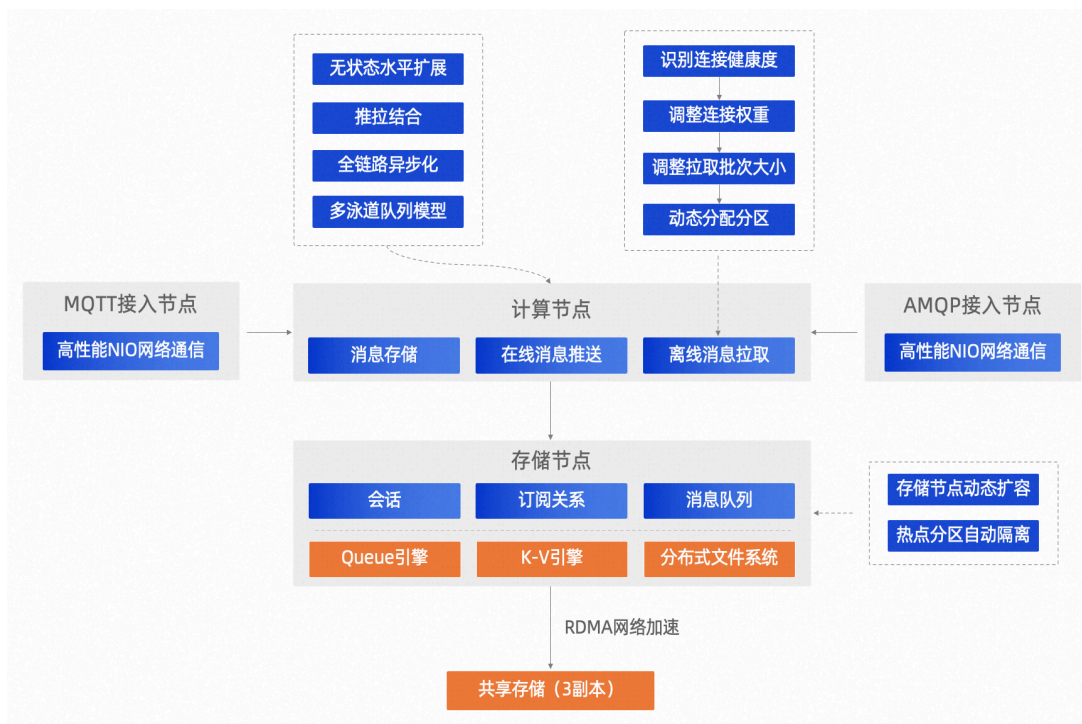
➤ 规则引擎

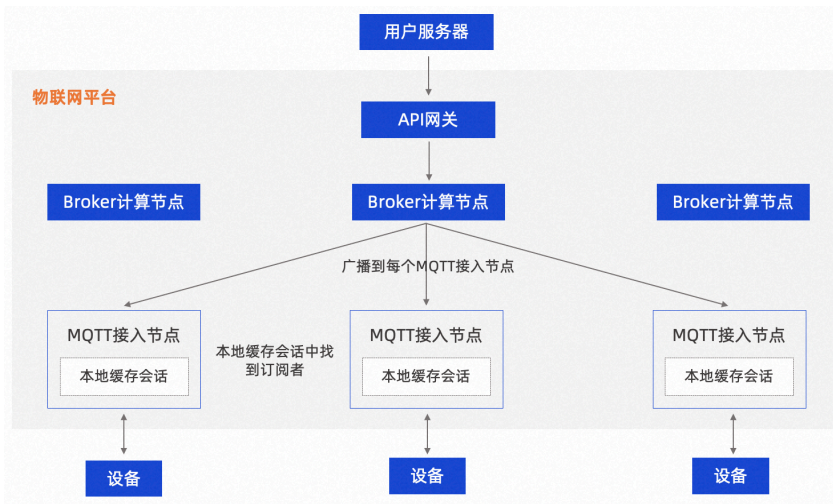
与传统的MQ不同，传统的MQ往往聚焦于异步解耦、削峰填谷等。IoT消息队列的规则引擎是一个十分重要的功能。对接入平台的设备设定规则，在条件满足所设定的规则后，平台会触发相应的动作来满足用户的业务。典型业务有消息流转、场景联动等。面对各种复杂的业务场景，规则引擎如何设计、如何做到高性能、如何具备扩展性、如何保障稳定性，都是规则引擎的技术挑战。

➤ 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





物联网平台消息队列采用连接、计算、存储分离的架构。接入节点包括MQTT接入节点和AMQP接入节点，支持千万设备长连接。计算节点是无状态的Broker，负责消息分发。存储节点负责存储会话、订阅关系和消息队列。

在这种架构下，各个节点职责清晰，可分别支持水平扩展。节点之间通过高性能NIO网络框架进行通信，并采用RDMA技术进行网络加速。通过以上连接、计算、存储分离的架构，做到了高性能、高可用，架构具有良好的弹性。

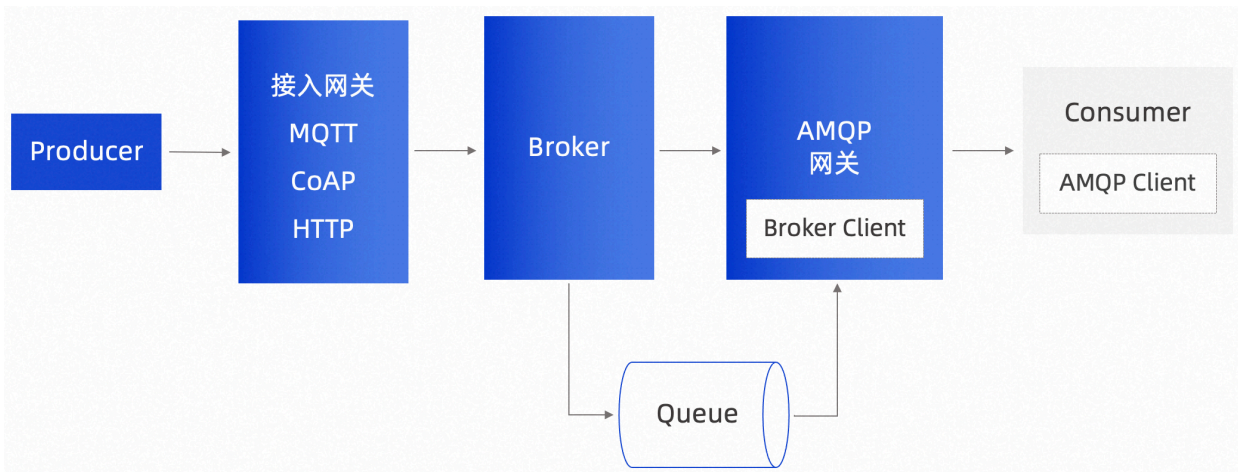
海量Topic：存储引擎升级

传统消息队列，其存储引擎简单来说是一种Queue引擎，在Topic数量变多时会产生大量的随机读写，严重影响性能。针对传统MQ的问题，结合IoT场景的业务特性，物联网平台升级了存储引擎，在Queue引擎的基础之上，自研了K-V引擎，支持海量Topic存储，同时支持海量会话和订阅关系的存储。

对于AMQP服务端队列，采用Queue引擎，达到极致的读写性能。

对于MQTT设备队列，采用K-V引擎，支持海量Topic。

实时优先：推拉结合



传统的MQ采用的拉模式，是传统的FIFO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。

秒级消息广播

- 突破传统的pub/sub:

如果查询某个Topic被百万设备订阅，查询过程将非常耗时，单个Broker节点计算能力有限。

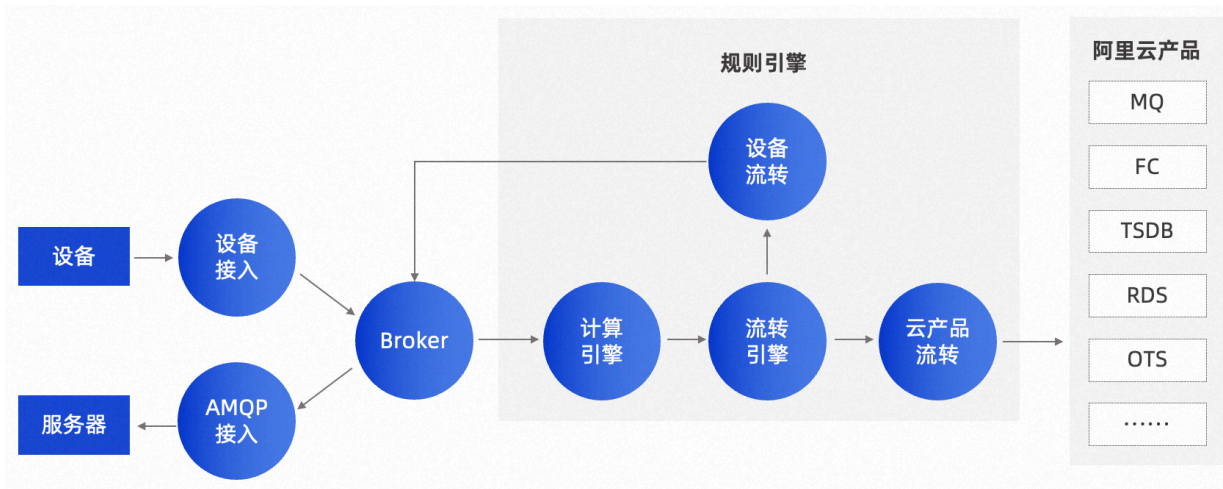
- 以空间换时间：

将设备订阅关系本地缓存到各个MQTT接入节点，避免去存储节点做非常耗时的查询。

- 分布式计算：

类似map-reduce，将广播消息广播到每个MQTT接入节点，接入节点内部查询本地缓存找到百万订阅者，然后进行消息推送，充分利用了多个MQTT接入点的算力。

➤ 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

- 高可用架构



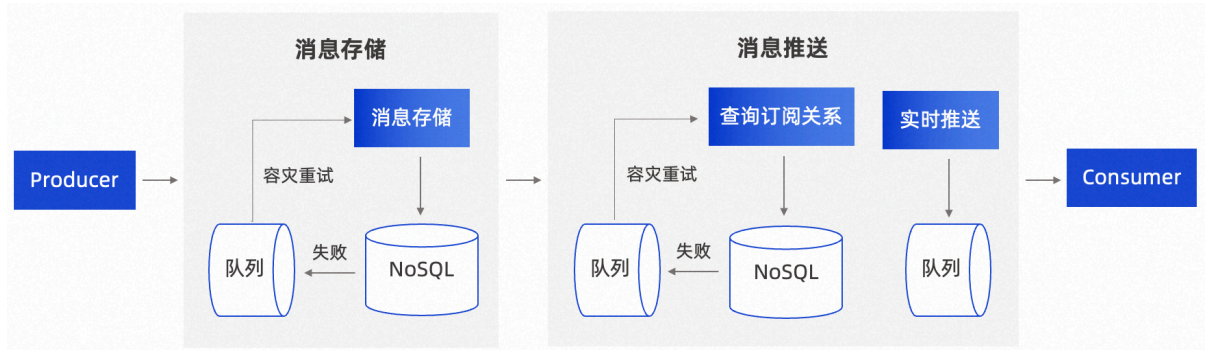
1) 接入节点高可用

- 连接健康度识别。
- 会话自检。
- 跨可用区容灾。

2) 计算节点高可用

- 秒级故障转移。

- 无状态水平扩展。
 - 跨可用区容灾。
- 3) 存储节点高可用
- 分布式文件系统+共享存储的设计，做到秒级故障转移。
 - 热点分区自动隔离，跨可用区容灾。
 - 数据3副本。



● 消息高可用

1) 容灾

目前消息的容灾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。

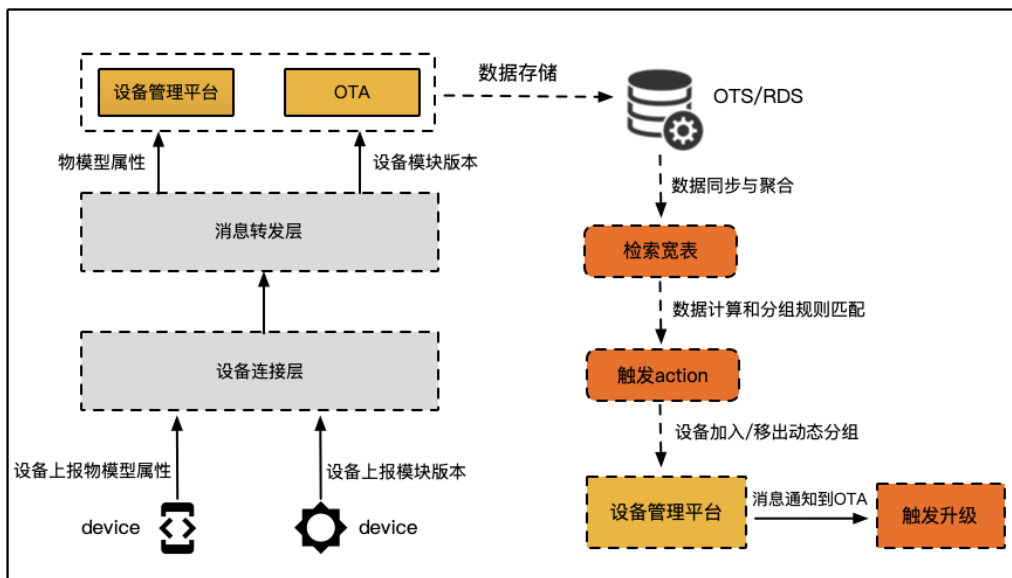
3.2.3. 核心技术点

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. 大规模设备管理

3.3.1. 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、低时延、灵活的查询，所面临的主要技术挑战包括海量数据存储、数据高频变更、多源异构数据聚合。

- 海量设备的管理运维

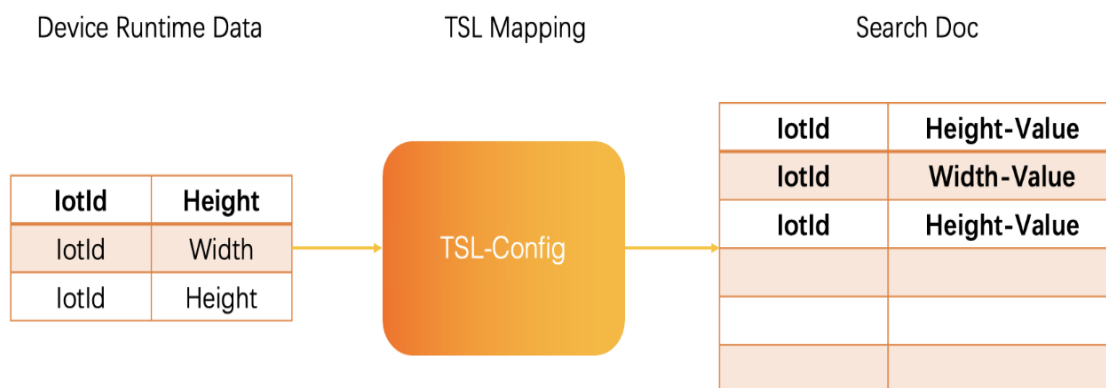
物联网场景下，连接上云的设备数越来越多，当设备规模达到亿级规模，如何对大规模设备进行管理和控制，例如大规模设备控制和远程升级，是一个非常大的挑战。涉及到设备圈选能力、设备任务调度能力、云端向设备批量推送能力，这些诉求对系统稳定性也提出了更高的要求。

3.3.2. 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态和静态两种能力。静态检索，是基于已有数据检索符合条件的设备，动态检索，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、动态分组等。

- 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举的。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。

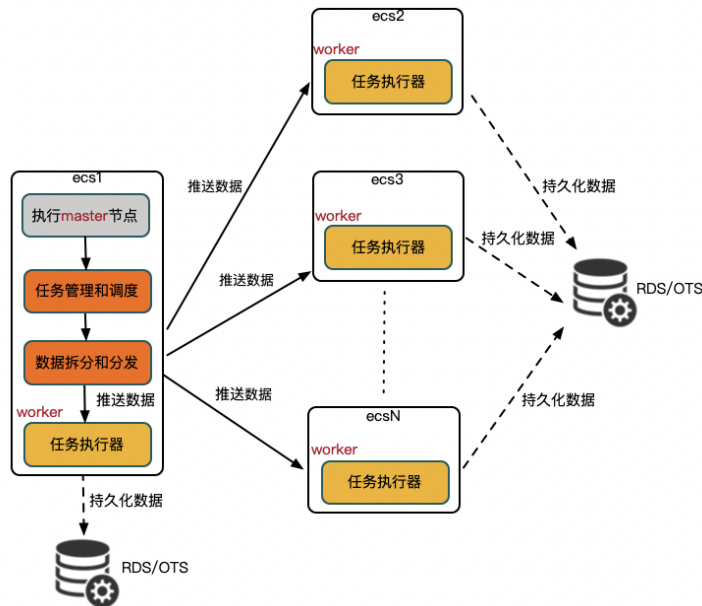
主要的技术手段是通过结合物模型数据定义明确、整体数量不可穷举、单设备可穷举的特点，将单设备的物模型信息与索引进行映射，多设备复用相同索引，实现物模型数据的检索。

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，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、DeviceName文件列表、灰度策略、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来发起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，还提供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批次下设备的并发升级。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。

3.3.3. 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。

3.4. 设备建模

3.4.1. 物模型

3.4.1.1. 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

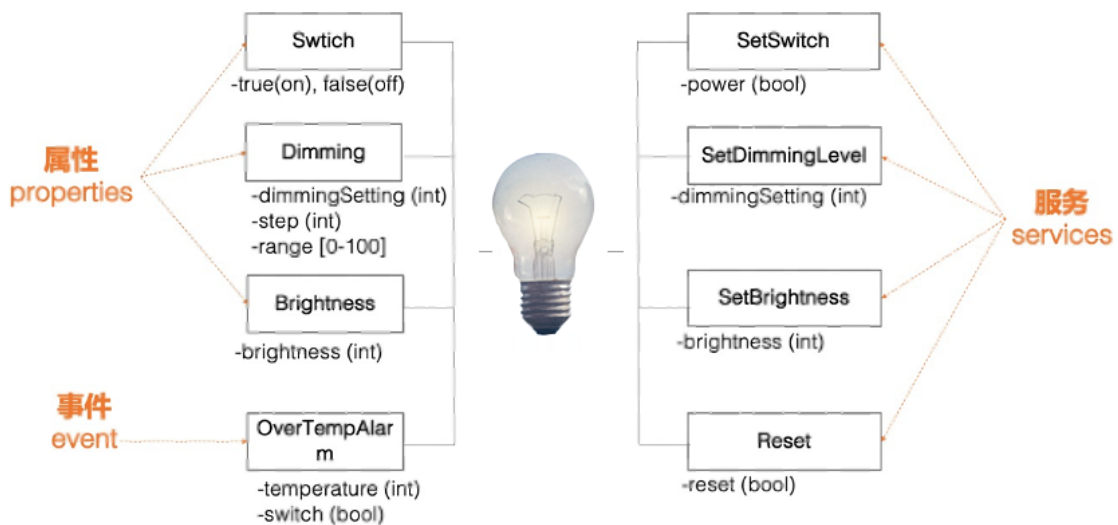
针对业务场景构建的复杂业务系统模型，每个数字孪生体支持万级节点，每个节点支持300个属性定义，每个物模型属性又可配置10条数据计算规则。物理设备数据映射到孪生节点之后，会自动触发多级节点间的数据运算。如何保障孪生体数据规则执行的效率和稳定性，为我们带来了很高的技术挑战。

3.4.1.2. 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。
- 2) 以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、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针对每种数据类型还定义了非常严谨的数据规范，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取值范围为1摄氏度至100摄氏度。

- 3) 同时，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，以帮助用户快速高效的完成建模工作。

随着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，能够描述多级设备间的树形关系结构，并支持配置节点以及节点间联动的规则运算，以实现设备间复杂的业务逻辑。

以某园区温度统计进行简单说明：

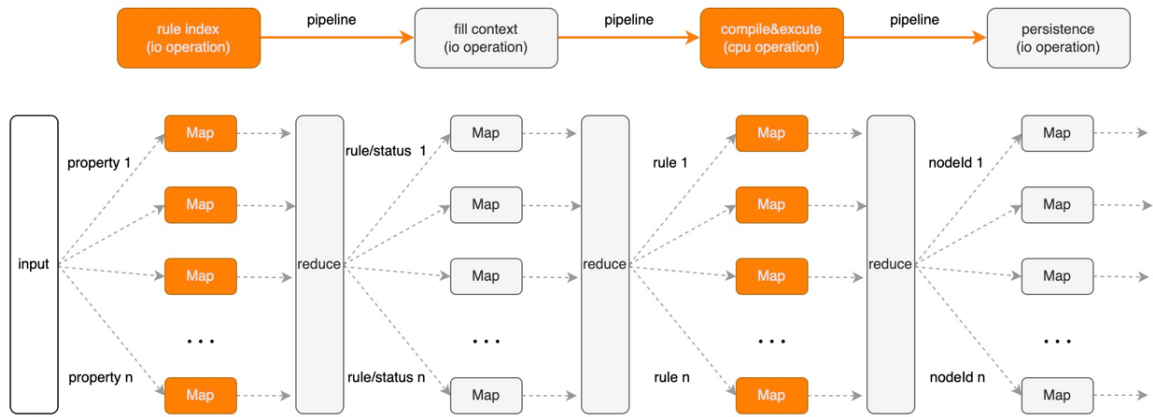


（使用图数据库构建的园区设备关系，通过温度传感器计算办公区平均温度。）

- 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、空间、虚拟实体，每个节点可定制物模型功能定义和规则。构建复杂业务场景多层次设备关系，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。
- 模型定义结束，下一步就是数据的采集，物理设备的数据采集可通过数据解析功能映射到孪生节点，实现设备数据和孪生业务模型的解耦。同时平台还支持API和离线表等多种数据源用于接入孪生。
- 原始设备的数据通常不足以满足业务场景，为了应对复杂的节点间计算，平台提供了脚本表达式，可在单节点或多节点间作计算和联动，以实现节点间的通信。
- 为了方便实用，孪生体提供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，以及孪生节点间关系检索。同时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。
- 在检索的基础上，不仅可以满足用户的查询诉求，还可以基于检索结果对设备发起批量控制。

➤ 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关联节点，再通过异步消息触发关联节点数据规则运算。通过对链路进行解耦，并借助运行态事件响应的手段，达到动态调配系统负载以提升执行效率的目的。
- 2) 为了避免规则执行出现环路。在管控态创建孪生规则时进行有向无环图的严格检查，以降低系统运行时风险，同时系统还有自检能力，只要发现此问题就会立即终止运行。
- 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，对配置规则进行预编译行，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。

3.4.1.3. 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	TSL2.0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。

3.4.2. 时序数据

3.4.2.1. 核心技术挑战

➤ 高效性的时序数据存储策略

IoT设备时刻在产生海量的时序数据，这些时序数据中部分数据具备较高价值需要进行存储，而部分数据价值较低并不需要进行存储，全量存储时序数据会造成大量的存储成本；同时时序数据具备时效性，随着时间的增长，部分高价值数据也会转换为低价值数据，需要及时清理；

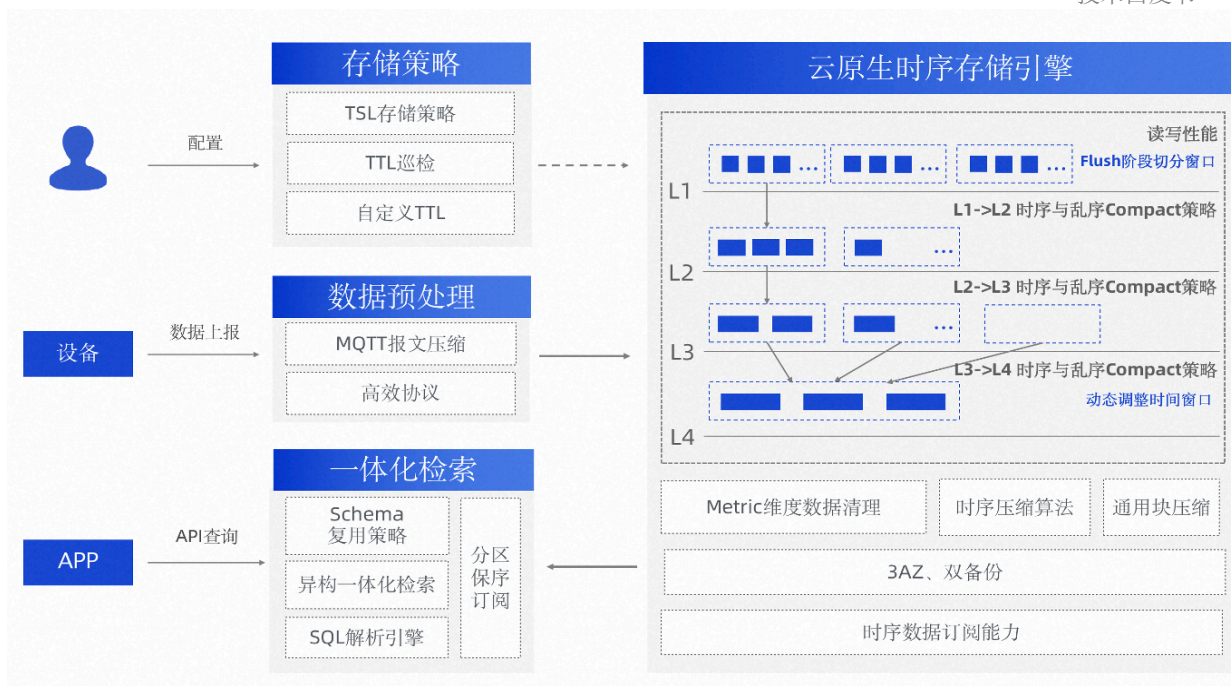
➤ 高效的统一时序存储模型

IoT设备时序数据具备海量、异构、超高写入QPS、高查询QPS、时间序列等特点。如何对海量异构的时序数据进行统一的组织，决定了能否提供通用的查询和写入能力、能否承载超高写入QPS和高查询QPS的流量、能否提供持续稳定的服务。

3.4.2.2. 技术详细描述

➤ 时序数据存储策略

为了降低时序数据的存储成本，除了高效与多维度的时序数据压缩算法外，物联网平台在产品层面还为用户提供了灵活的时序数据存储策略与时序数据TTL策略（Time To Live）。在数据存储前，可自定义地对低价值数据进行过滤，只存储高价值数据；在数据存储后，对于过期数据进行清理与冷备，提升存储空间利用率。



➤ 时序存储模型

高效的存储模型能够极大地提升时序数据的写入与查询性能，同时也能提升稳定性。物联网平台结合前述物模型，将海量的时序数据抽象为统一的存储模型，能够满足不同场景下通用的的查询方式，同时基于统一的存储模型对于写入与查询进行统一的优化，极大地提升了时序数据写入与查询的性能与稳定性。

Table	lotId	TSL			Timestamp
		Property	Event	Service	
PK/TSL	MxizS120JTkWOeqXdt10kObox0	Temperature	Alarm	Switch	2023-10-13 17:53:25.531
	MxizS120JTkWOeqXdt10kObox0	1	True	on	2023-10-13 17:53:26.531
	MxizS120JTkWOeqXdt10kObox0	2	False	off	2023-10-13 17:53:27.531
	MxizS120JTkWOeqXdt10kObox0	2023-10-13 17:53:28.531
	MxizS120JTkWOeqXdt10kObox0	4	True	off	2023-10-13 17:53:30.531
	MxizS120JTkWOeqXdt10kObox0	5	False	on	2023-10-13 17:53:30.531
	MxizS120JTkWOeqXdt10kObox0	7	True	off	2023-10-13 17:53:30.531

Metric | **Tags (数据来源)** | **fields (数据内容)** | **Timestamp (数据生产时间)**

3.4.2.3. 技术核心点

技术	说明
读性能	支持40万级点位实时上报，云原生架构，支持弹性伸缩，读写性能可线性扩展。
写性能	支持万级点位时序数据实时查询，云原生架构，支持弹性伸缩，读写性能可线性扩展。
多维度存储策略	万级TTL策略、时序存储策略、检索Schema复用策略。
数据压缩	时序存储压缩率最高可达10:1。

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。

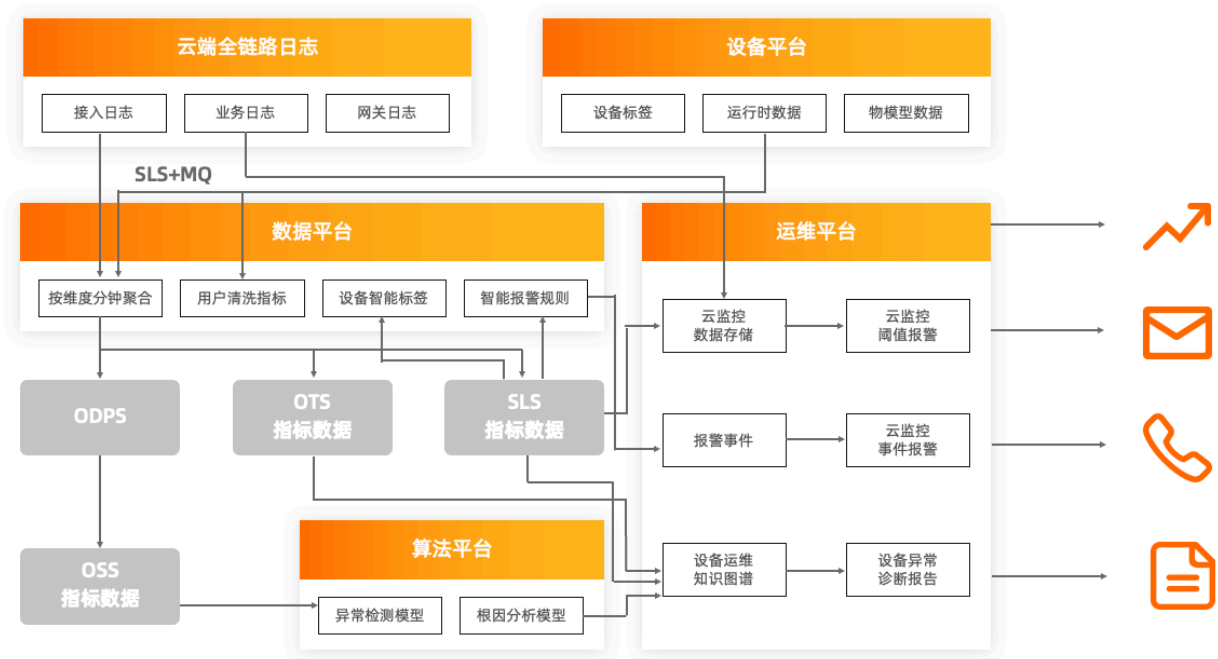
► PB级日志数据分析处理

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例如实时在线设备、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，帮助客户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。

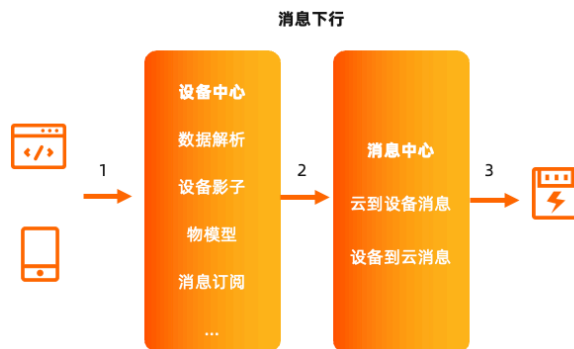
➤ 消息轨迹

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：

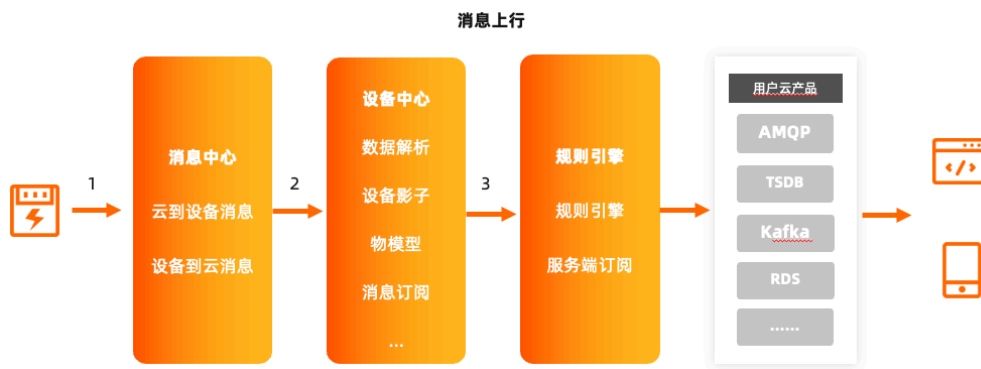
1) 链路业务抽象

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. 数据服务

3.6.1. 核心技术挑战

➤ 多样化数据格式的解析支持

在使用物联网设备时，不同行业的设备数据具有不同的数据格式。如何设计一套工具，兼容不同数据格式的解析需求，把碎片化的设备上报数据格式，转化为标准的可供业务人员分析理解的格式化数据，这是数据解析功能面临的一大难题。与此同时，常见的实时数据处理方案具有较高的成本门槛，在低频设备数据消息的场景下，会给用户带来过高的成本负担。

➤ 丰富使用场景下海量设备的数据存储

物联网设备数据规模巨大，常见的单个关系型数据库并不能承载如此大量的数据，此时需要基于服务器集群构建可以承载海量数据的存储底座。另一方面，在形如设备实时控制的场景中，需要利用到近期的时序数据，需要考虑到数据系统的高并发、低延时与极高的稳定性；而在对设备历史数据进行分析或对未来进行预测的场景中，需要利用到过往的海量数据，此时需要考虑的是数据存储系统能不能完成海量数据的压缩存储与复杂的数据分析。因此，如何为不同业务使用场景的用户，提供多样化的存储方式和统一的数据生命周期支持，并保证系统的可扩展性、稳定性、低成本等因素，是数据存储系统的核心挑战。

➤ 长时间周期的海量设备数据价值的挖掘

物联网设备上传的数据信息密度相对稀疏，要从中获取有价值的业务洞察，需要对历史中长周期的海量数据进行分析，稀疏的海量数据带来高计算分析成本。如何提供相对较低成本的海量数据的分析能力，是物联网平台数据服务面临的核心挑战。同时为满足用户低门槛挖掘分析数据，平台需要提供与用户场景紧密结合的统计算子与AI算子，让用户可以通过简单的产品交互就满足深度分析洞察的诉求。因此如何设计简单易用的统计算子与AI算子，让用户能够通过一个平台完成基础的数据分析和进阶的AI算法分析，成为要解决的重要问题。与此同时需要充分考虑服务的可扩展性、低成本等因素。

3.6.2. 技术详细描述

➤ 数据解析

数据解析模块对接入数据服务平台的数据提供实时处理能力，通过数据解析模块，可以完成对设备上报的数据的格式解析和预处理。解析得到的结果可以以相对较低的成本存储在数据服务平台的存储模块中，或通过规则引擎流转到其他下游模块进行进一步的处理。通过以下几项技术，可以在保证低成本的同时，完成对多样化数据的支持：

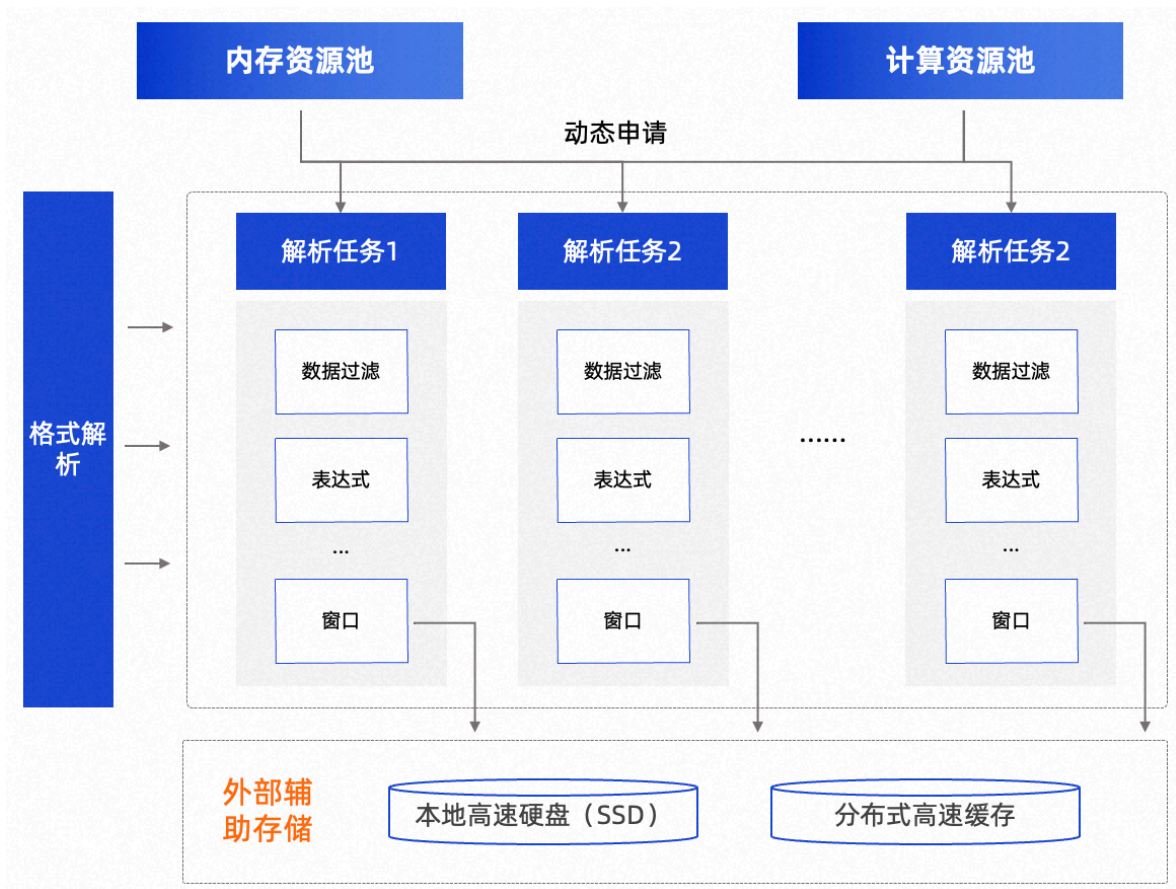
1) 灵活的数据格式支持，支持预置和自定义数据格式

如下图所示，数据解析可以对上传到平台中的多种消息格式进行解析，除了JSON数据格式外，还提供了ProtoBuf、Base64两种常用的编码格式解析。对于碎片化的物联网数据格式，需要提供一种灵活且通用的解析机制，能够让用户自定义解析逻辑，实现不可预知的用户数据格式解析需求。数据解析基于阿里云函数计算（Function Compute）的能力，支持用户使用JavaScript、Python和PHP三种语言来编写解析脚本，灵活的对上传的数据格式进行解析。



2) 池化计算资源

通过动态加载及计算、存储资源池化的方式，硬件资源可以得到极致的利用、用户体验更加流畅。如下图所示，对于需要进行解析的消息，模块首先会将其对应的数据解析逻辑加载到内存中，并在运行时动态申请计算与存储资源。该数据解析逻辑在一段时间未使用后，会自动释放资源，并序列化到高速缓存中，等待再次被加载。



对运行时用到的算子进行分类存储，可以有效降低内存占用。每个解析任务在配置序列化后可以做到只占用 10 KB左右的内存。对于双流JOIN、维度扩展、将采样和移动平均等内存不友好的算子，我们将其存储在外部的辅助存储中，在调用时拉取到内存内；对于表达式、数据过滤和条件转化等算子，我们将其保留在内存之中。通过这样的方式，可以完成以极低成本启动解析任务，单机加载海量解析任务的挑战。

➤ 数据存储

数据存储旨在为丰富使用场景下的海量物联网数据提供多维度、多样化和低成本的存储服务。支持客户按照不同的使用维度，不同的使用场景来构建属于自己的数据存储仓库。为了方便客户的使用，数据存储同时提供多种数据源的接入方式，以满足客户数据集成的需求。

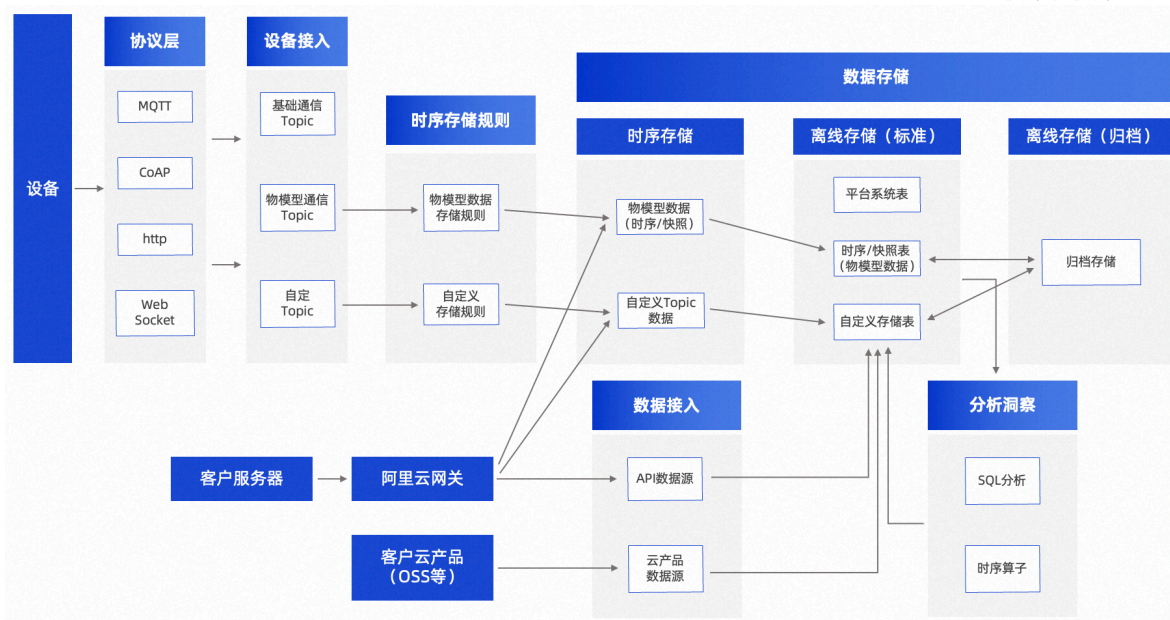
1) 满足不同场景的数据存储

按照存储方式的不同，为客户提供时序存储、离线存储（标准）和离线存储（归档）三类存储，以便满足用户对物联网平台数据不同使用场景的需要：

高并发、低延时的时序存储：物联网平台数据服务下时序存储的目标是协助用户搭建物联网场景下低延时与高并发处理设备数据的存储底座。通过采用TSDB数据库，满足了用户对设备时序数据高并发、低延时读写的需要。

支持海量数据SQL分析的离线存储（标准）：数据服务下的离线存储基于阿里云盘古分布式存储系统构建，提供最大可达EB级别的存储。离线存储中的标准存储采用三个副本存储模式，为数据的可靠性提供保障。

低成本的离线存储（归档）：归档存储是一种针对不经常访问但需要长期保存的数据的存储形式。归档存储旨在通过较低的存储成本为企业长期数据保留和合规性支持服务。对于不需要实时读取，且短时间内暂无SQL分析需求的历史数据，用户可以采用将数据进行归档的方式来降低存储成本。



2) 多种数据来源

如上图所示，数据存储中数据来源可以分为平台接入的设备数据源、外部自定义数据源两大类：

- 通过物联网平台接入的设备，上报的设备数据在经由时序存储规则模块处理后，可以存储在平台提供的时序存储系统中，用于满足用户业务系统实时高并发查询的要求。同时平台提供将此类设备数据备份到离线（标准）存储的能力，那些无需长周期存放在时序存储中的数据可以持久保留在平台的离线存储中，以供后续对长周期内产生的海量设备数据信息进行关联分析。
- 通过其他方式收集到的物联网设备或其他数据，也可以通过平台提供的API源导入、OSS同步或表格文件上传方式将数据导入到平台提供的离线存储系统中，以供后续的进一步分析使用。

3) 支持系统与自定义维度数据

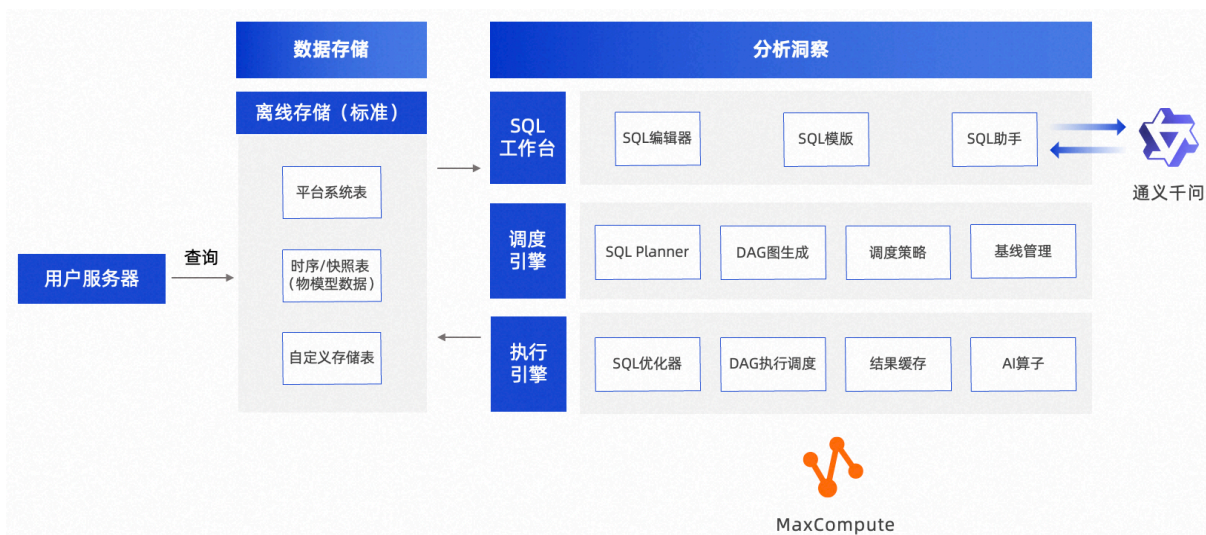
在数据仓库概念中，除了设备上报的大量原始数据（实时数据）外，还经常需要用到更多设备相关的维度数据，例如用户需要对某个厂房的一批设备进行汇总分析的场景中，需要通过SQL使用JOIN语句与厂房设备关联关系表进行关联查询。

对于直接接入到阿里云物联网平台的设备，数据存储中已内置了相关的维度表，包括产品表、设备表、设备分组关系表等，用户可以在分析洞察模块中基于这些维度表完成与设备数据的关联分析。

如果有自定义维度表的需求，用户可以通过调用API数据源功能透出的阿里云开放API接口，将相关维度数据写入到自定义存储表中，以供后续使用。

➤ 分析洞察

物联网平台中分析洞察向用户提供从长时间周期的海量数据中挖掘设备数据价值的功能。分析洞察是离线数据处理模块，支持以天或以小时为周期定时执行计算任务。用户通过SQL语句编写计算任务，SQL语句中可以对数据存储中的各类表进行复杂关联查询，并应用相关的算子，充分挖掘物联网设备数据中的价值。分析洞察任务周期执行后，将结果



写入到自定义存储表中。用户通过阿里云开放API接口查询结果。下面是分析洞察模块的整体架构：

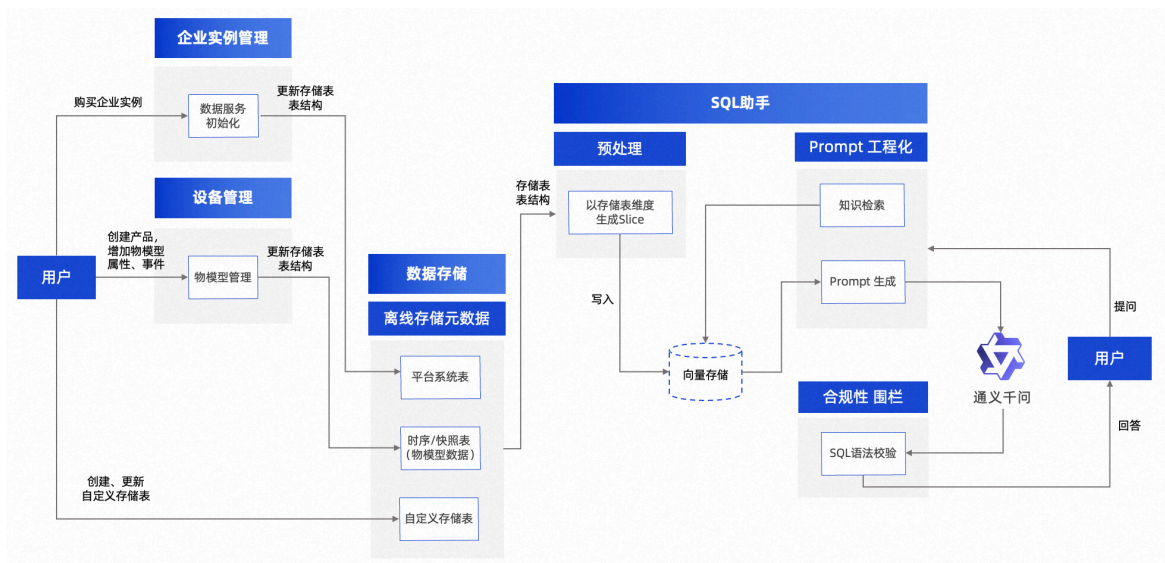
分析洞察能力的核心是支撑对用户存储的海量物联网数据的联表关联查询，以及通过AI算子对数据进行深度的分析与洞察。这里主要通过以下几个关键技术，解决分析洞察遇到的海量数据分析的使用成本、易用性等方面的技术挑战：

1) 构建低成本的海量数据分析能力

物联网平台基于阿里云的MaxCompute大数据产品构建了海量设备数据分析洞察的能力。调度引擎在生成调度计划时，基于每个任务历史上执行时长与执行成本，生成了最优的调度计划，最大限度利用了MaxCompute计算资源，以降低数据洞察任务成本。执行引擎中的SQL优化器通过谓词下推的技术改写SQL，可以降低SQL读取的数据量，缩短SQL执行时间，降低SQL执行成本。此外构建了资源水位监控能力，在资源水位不足的情况下通过自动扩容的手段增加系统计算资源，以避免出现因资源不足而造成任务无法被执行的问题。

2) 通过SQL助手快速生成分析任务

分析洞察提供的SQL助手借助通义千问大模型，为用户提供自然语言转SQL的能力，用户可以通过自然语言提示词快速产生SQL。此外，分析洞察提供了常用的SQL模版，当中包含了常见场景下SQL语句片段。借助SQL助手与SQL模版，可以有效降低用户创建分析洞察任务的门槛，提高用户使用效率。SQL助手的执行过程示意图如下：



SQL助手的执行过程主要分为两部分：

- 预处理：将用户表结构信息存储在向量存储中。
- 用户提问与回答：用户提问时从向量存储中检索出相关信息，生成提示词。通过通义千问获得结果后，经过合规性围栏的检查，将结果向用户展示。

3) 通过简单易用的算子提高分析效率

物联网平台分析洞察模块除了提供大量标准的SQL函数以外，同时也提供了常用的机器学习等AI算子，用户可以将数据处理过程和AI算法过程结合起来，统一构建自己的AI数据分析场景。会话级别的沙箱隔离，可以保证每个自定义函数单独运行的稳定性，避免了不同AI算子之间的库依赖冲突问题，给予用户更稳定的使用体验。示意图如下：



3.6.3. 核心技术点

技术	说明
数据解析	通过预置的JSON, ProtoBuf等格式支持, 同时集成了函数计算能力, 实现了对用户多种协议的支持。通过创新的资源管理方式, 实现了细粒度的计算资源管理。用户可以用更低的使用成本门槛来创建一个实时数据解析任务, 处理低频设备数据消息更加友好。
数据存储	提供时序存储与离线存储功能, 可以满足用户时序计算与离线分析两种应用场景的使用要求。同时提供了归档存储来存储需要长期保存但不需要实时读取或分析的数据, 以进一步降低存储成本。并提供了丰富的数据接入, 和物联网数据源、文件导出、数据Open API三种形式的系统集成服务。
分析洞察	物联网平台基于阿里云的MaxCompute大数据产品构建了海量设备数据分析洞察的能力。通过整合通义千问大模型, 为用户提供了自然语言转SQL的能力。用户可以用SQL函数的形式使用AI算子, AI算子执行时采用沙箱机制, 不同算法实现之间互不影响。

3.7. 数据智能

3.7.1. 核心技术挑战

► 时序异常检测

- 1) 随着联网设备的不断增长，运维的难度也在不断增加。针对大规模的物联网设备集群，传统的基于人力的运维方式，无论是运维的准确性，还是成本都是让人难以接受的。一些近年来提出的异常检测算法希望通过模型来替代人力运维，但这些基于统计算法的模型，往往需要一个设备一个模型，当设备数量较大时，会带来极高的计算开销。并且由于计算是离线进行的，难以对数据进行流式处理，得到的检测结果也往往是延后的。如何针对物联网场景，提供实时的，低成本的异常检测算法，降低设备运维难度，是非常重要的问题。
- 2) 传统的异常检测算法，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。这对异常检测算法的设计是一个不小的挑战。
- 3) 设备状态会由于内部、外部因素随着时间而改变，各类时序指标的波动形态可以间接反映设备状态随时间的变化。例如：
 - 在设备运转过程中，电机转速在某一时刻可能由于故障而突然变大或变小。
 - 在润滑油液监测过程中，油液的含水量可能由于设备进水发生明显增长。
 - 在生产过程控制中，产品质量可能由于加工设备的故障发生偏差。

这些问题都会导致相应监测指标的波动形态发生变化，且不同场景机理下指标波形变化可能千差万别。如何准确且及时地检测出时序的波动变化以反映潜在事件也尤为重要。

► 光伏阵列异常检测

太阳能发电对于缓解能源短缺以及减少对化石能源的依赖具有非常重要的作用。近年来，太阳能发电产业始终保持着较快增长。

- 1) 由于光伏阵列常年暴露在户外，受多变环境（例如雷雨天气、热循环、湿度、紫外线辐射、阴影等）影响，容易出现局部材料老化、裂纹、热斑、开路或者短路等各种故障，影响其使用寿命。
- 2) 集中式光伏电站中设备安装量大，但由于地处偏远地区，难以投入足够的人力、物力对电站的光伏设备进行定期全面的检查。如果不能及时对产生的故障进行诊断并处理，会显著降低光伏阵列的发电效率，且可能造成火灾等严重后果。
- 3) 分布式光伏安装分散，更加难以保证足够的人力进行后期维护，潜在的设备故障极大的降低了光伏发电的经济价值。

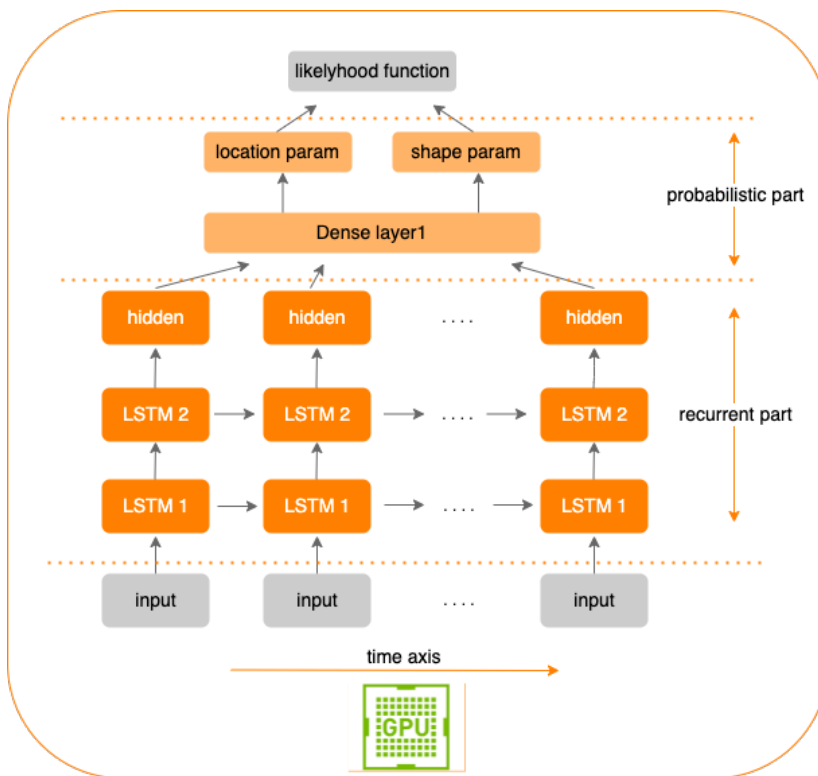
3.7.2. 技术详细描述

► 时序异常检测

物联网平台支持的时序异常检测功能包括：点级别异常检测和子序列级别异常检测。

1) 点级别异常检测

该功能主要解决时序数据上下文单点异常，所采用的算法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”。如下图所示：



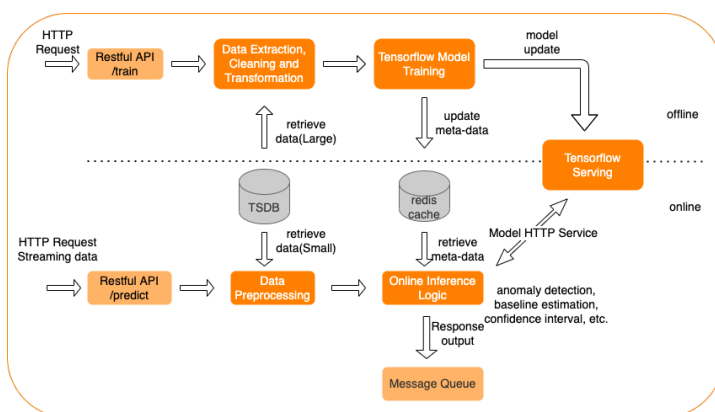
算法框架主要包含三个主要步骤:

- Step1: 利用循环神经网络LSTM对历史时序数据进行特征提取。
- Step2: 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。
- Step3: 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。

这里主要通过以下几个关键技术, 解决物联网场景下实时异常检测遇到的挑战:

- 海量设备的实时流式检测

为了应对海量设备的高性能检测的挑战, 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。

- 为时序预测算法赋予时序检测能力

在应对算法开发的高准确率技术挑战中, 我们在原始论文的高斯对数似然损失函数的基础上, 引入了基于伯努利

似然的损失函数。

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，例如1%、3%、5%等级，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。

- 非平稳时序数据的建模

应对物联网非平稳数据的挑战中，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。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。

- 降低误报率

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：

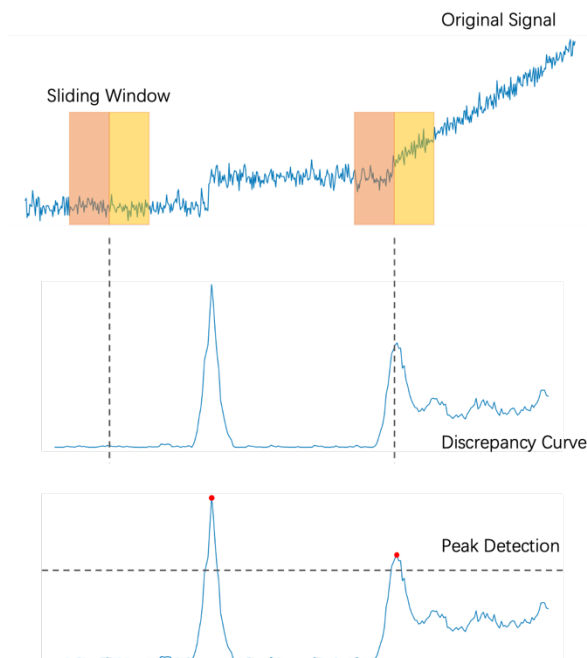
- 模型自身参数：例如层数，神经元数。
- 训练过程参数：批大小。
- 防止过拟合参数：dropout, regularization等。

训练区间为过去14天到过过去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。

2) 子序列级别异常检测

该功能主要解决时序数据集异常，算法采用基于滑动窗和特征损失函数的快速估计方法，支持离线和准实时在线检测，可以检测多种序列特征偏离场景。由于数据形态发生偏离的时间点简称变点，此算法又称为变点检测。

假设某指标的时序数据形态变化大致呈现如下图所示的三个阶段：第一个阶段在较低的区间波动；第二个阶段由于背后某事件的发生，指标跃变到稍高的水平波动；第三个阶段由于潜在事件（例如裂化加剧等），指标开始逐步波动升高。两次事件的发生导致了该指标呈现前后三种不同的波动形态，其中，分数差异曲线的两个峰值点即对应了两次事件的发生：



图中方框为一次滑动窗口，包括前后两个子窗口，通过不断计算相邻窗口间的时序数据的差异获得分数曲线，找到差异分数最高的若干点并结合一定的惩罚阈值可以按需筛选出变化最大的点，从而区分不同波动特征的数据形态。

算法主要步骤如下：

- Step1: 根据检测信号长度和窗口尺寸初始化分数向量。
- Step2: 从信号起点开始按照指定步长滑动窗口，直至终点，每次滑窗，利用代价函数计算相邻窗口时序数据的差异，并赋值相应位置的分数向量；如果相邻窗口数据波动形态发生较大变化，差异就会随之变大。
- Step3: 全部滑窗计算完毕后，通过对更新的分数向量进行峰值检测获得若干极值点。
- Step4: 根据极值点和惩罚阈值筛选出最终变点结果。

算法主要通过以下几个技术组合，解决设备指标时序集合异常检测中遇到的挑战：

- 灵活适应多种场景

不同场景下设备机理不同，并且受工况、环境等影响，指标时序数据的波动形态可能千差万别，通常会发生多种特征的复合偏离变化，算法设计并支持多种代价函数，以覆盖多种常见特征偏离，例如：

均值偏离：

$$c(y_{a..b}) = \sum_{t=a}^{b-1} \|y_t - \bar{y}\|_2^2$$

方差偏离：

$$c(y_{a..b}) = \frac{1}{b-a} \sum_{t=a}^{b-1} \|y_t - \bar{y}\|_2^2$$

趋势偏离：

$$c(y_{a..b}) := \sum_{t=a}^{b-1} \left\| y_t - y_{a-1} - \frac{t-a+1}{b-a} (y_{b-1} - y_{a-1}) \right\|^2$$

其中， $y_{a..b}$ 表示目标序列

此外，我们也支持自定义代价函数以应对各类复杂多变的客户场景，基于具体时序特征帮助客户设计更加匹配的代价函数，优化检测效果。

- 满足检测灵敏度和时效性

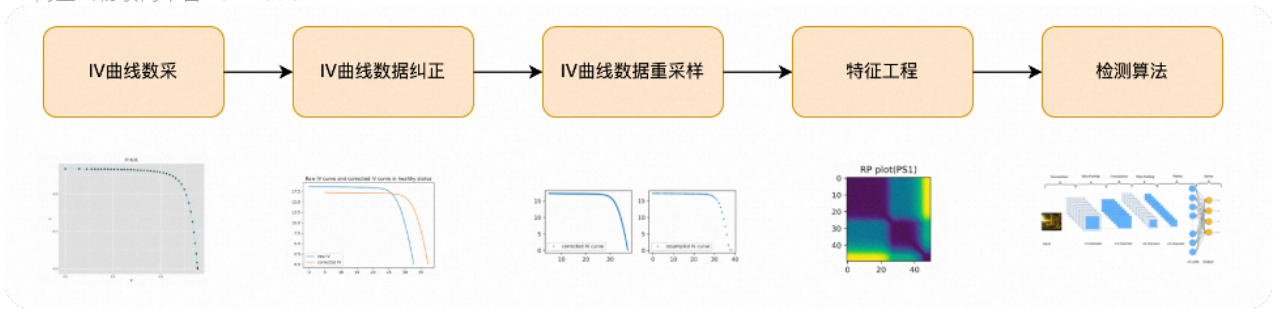
时序集合异常通常需要积累一定数据量，形成特定波动形态后才能判断是否存在形态上的异常偏离，天然存在滞后性，主要采用定时自动检测和离线人工调用。除了满足检测精度，某些场景对算法时效性也有不同的要求，如设备某指标劣化，需要在劣化确认发生但尚未造成严重后果前及时报警，由于场景灵活多变，算法设置了丰富的可配置参数，包括：

- 检测周期：设置越小可以提高检测时效性。
- 单次检测时长和滑动窗口大小：可根据检测周期自动计算理论最小值；也可自定义，配合检测周期优化检测精度。
- 滑动窗口步长：设置越大可以提高检测速度；设置越小可以提高检测灵敏度。
- 惩罚函数：应对不同场景下各种设备指标的不同量纲。

此外，算法还支持数据预处理（例如分钟级数据聚合）和结果后处理（例如相近时间变点去重），以满足不同时序粒度和检测粒度的场景需求，并降低误报。

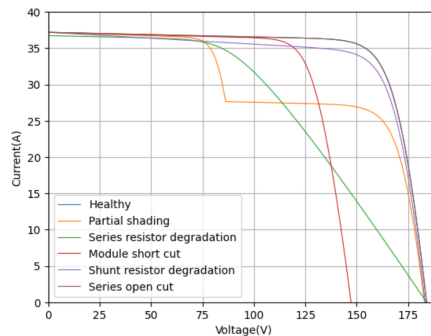
➤ 光伏阵列异常检测

阿里云物联网平台设计了基于光伏IV曲线分析的光伏故障诊断模型，通过采集光伏阵列的IV曲线，并结合机器学习算法，充分挖掘和学习不同状态下的IV曲线所包含的丰富信息，识别出光伏阵列中的故障，触发告警，提醒电站的运维人员及时排查解决问题。算法主要的运行流程如下所示：



3) 多种故障建模

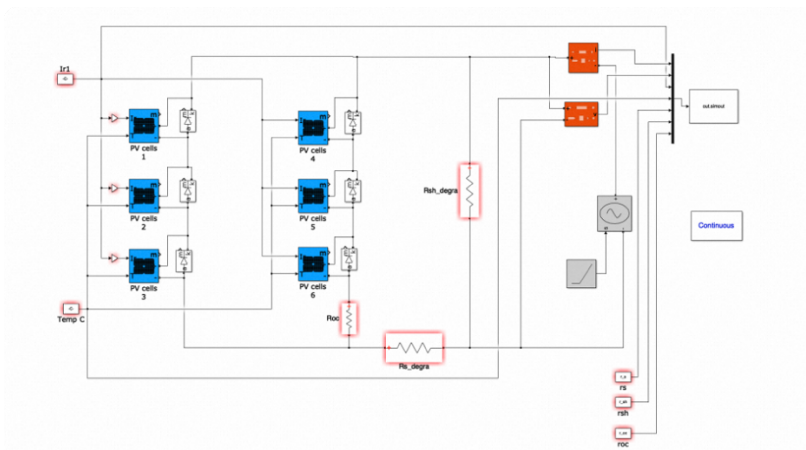
我们针对光伏发电中5种常见的故障进行了针对性的算法检测，故障类型包括：光伏阵列部分遮挡、光伏阵列串联电阻老化、光伏阵列并联电阻老化、光伏阵列组串开路、光伏阵列模组短路。当出现不同故障时，电流和电压呈现不同的非线性关系，如下图所示：



硅半导体材质的光伏电池，其IV特性不遵从经典的线性欧姆定律，检测算法最大程度从数据中抓取I-V之间的非线性特征，进而指导光伏组件故障的识别。

4) 光伏故障仿真

在实际生产中，故障数据的获取往往较为困难，数据的稀缺对算法的应用造成了极大挑战。为此，检测算法在利用已有的真实数据基础上，基于仿真技术进行光伏阵列的故障仿真，通过复现故障机理获取更加充分的故障数据，增强模型的效果。如下所示为一个光伏阵列的仿真等效电路设计图。



5) 数据处理

原始故障数据的外界照度和温度各不相同，需要对原始数据进行修正，映射到STC标准状态下，在统一的尺度下进行故障的检测。我们参照国际电工协会标准IEC 60891进行STC状态转换。转换方程如下所示：

$$I_2 = I_1(1 + \alpha_{rel}(T_{m2} - T_{m1}))G_2/G_1$$

$$V_2 = V_1 + V_{OC1}[\beta_{rel}(T_{m2} - T_{m1}) + a \cdot \ln(G_2/G_1)] - R_s(I_2 - I_1) - \kappa \cdot I_2(T_{m2} - T_{m1})$$

因采样工具和外界环境的不同，采样得到I-V的电压值并非均匀排列，I-V曲线的样本点数也不尽相同。为提高算

法的鲁棒性，我们采用最近邻法对原始的IV曲线进行重采样，保证电压的均匀分布和样本总数一致。

6) 机器学习建模

利用树模型对采集到的数据进行隐式特征提取，并利用集成学习降低模型过拟合。为此我们选用XgBoost对处理后的数据进行建模得到最终的检测模型。

3.7.3. 核心技术点

技术	说明
时序异常检测	点级别异常检测算法提供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。 子序列级别的变点检测算法适用于多种业务场景（包括但不限于：过程控制、质量管理、活动识别、监控维护），不同场景支持参数自定义灵活适配。
光伏阵列故障诊断	诊断算法在利用已有的真实数据基础上，基于仿真技术进行光伏阵列的故障仿真，通过复现故障机理获取更加充分的故障数据，增强模型的效果。

04 全球化

4.1. 重要性

近年来，中国企业在海外的市场机会越来越大，对海外数据合规的要求也越来越严格，也对设备连接速度和数据上云效率提出了更高要求。随着全球化和物联网技术的普及，物联网平台全球化的业务支撑变得越来越重要。

4.2. 挑战性

物联网业务实现全球化，需要依托于物联网平台的全球化部署、设备全球化就近接入、安全合规情况下的数据迁移、国际数据合规认证、设备全球连接速度等。这里面各项都需要进行大量人力、资源、技术的投入。核心需要解决的难题在于：

➤ 低成本全球化部署

将物联网平台低成本部署到阿里云全球所有站点，需要在有客户需求时即开即用。

➤ 设备全球售卖

大量设备在出货时并不知道会售往哪里，因此设备可以自动就近接入的能力就至关重要，需要做到设备可以在全球任意地方售卖和激活，并且保障数据合规。

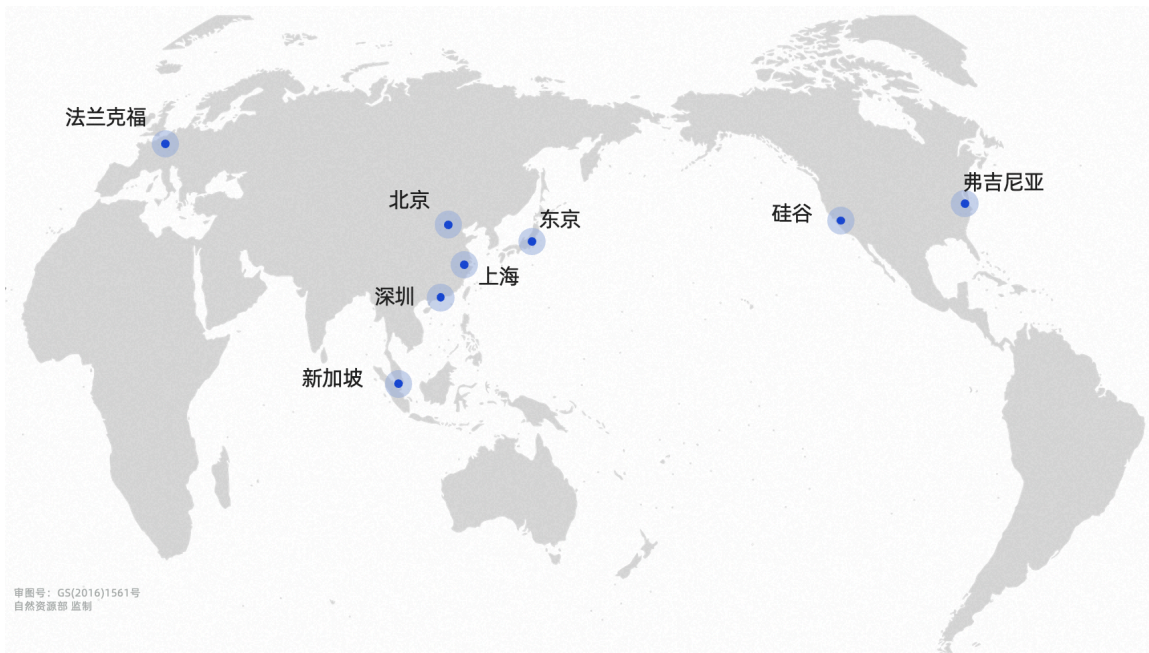
➤ 低延时设备连接

需要保障设备的连接速度，做到无论设备在哪里都可以以最优路径连接到云端，实现全球毫秒级接入。

4.3. 核心思路

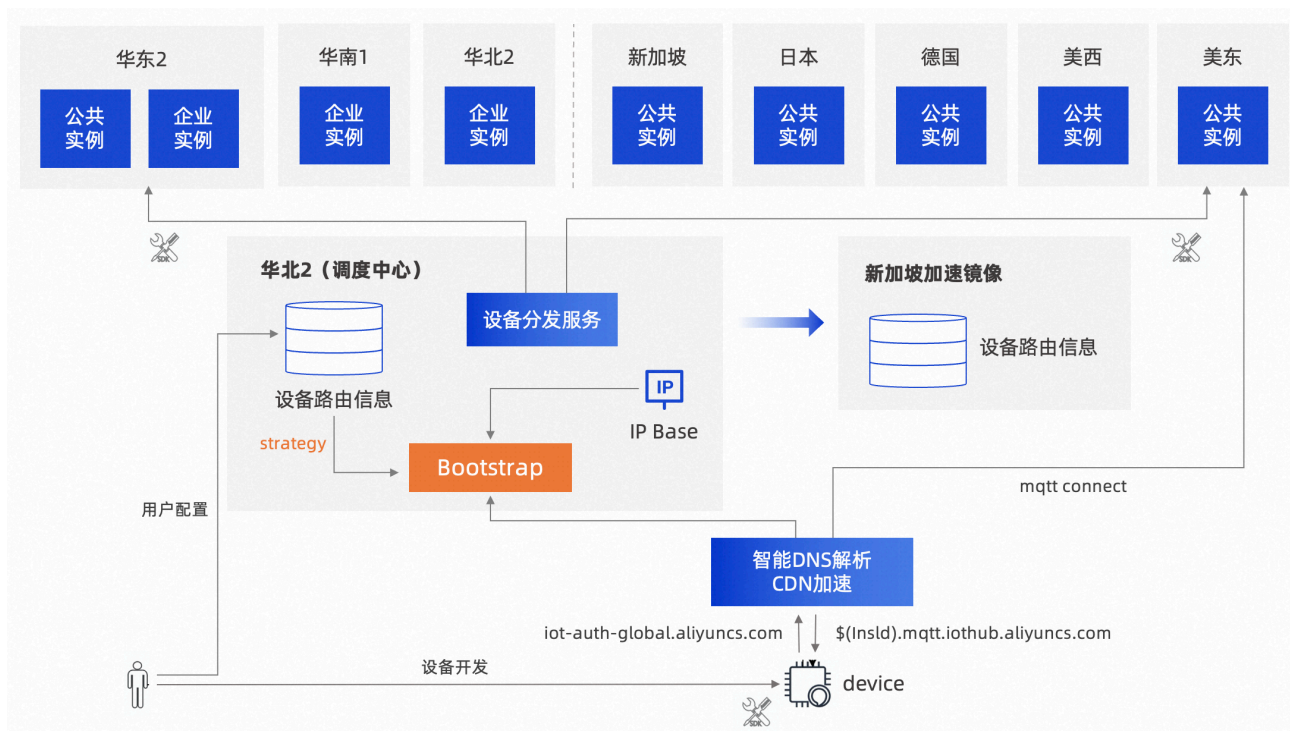
4.3.1. 全球化部署

物联网平台覆盖全球主要经济区，围绕中国企业出海业务活跃地区，全球部署8个站点，包括上海、北京、深圳、新加坡、东京、弗吉尼亚、硅谷和法兰克福，并在持续扩展中。



4.3.2. 设备全球就近接入

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Region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：



➤ 网络互通

全球就近接入服务支持设备全球范围生产，不同地域接入，在接入激活后，支持设备身份数据跨Region分发，跨地域通过跨域专线实现。

➤ 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。

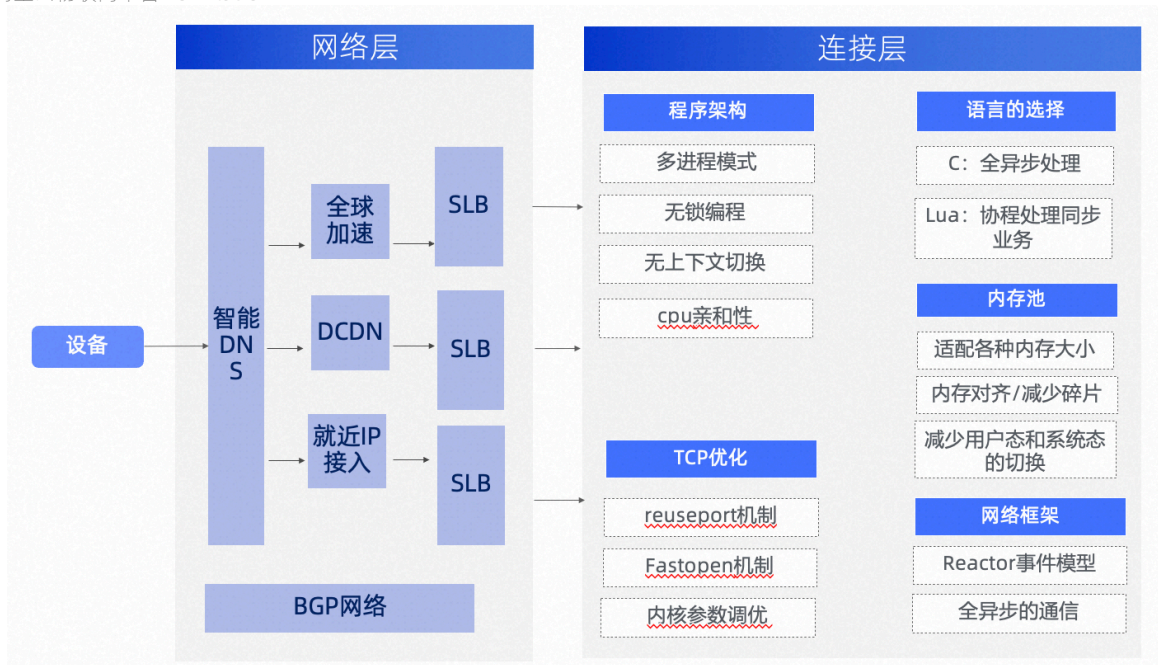
➤ 分发策略

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，来解决设备跨账号转移和全球就近接入的问题。

➤ 就近接入

设备只需烧录全球统一接入点，请求Bootstrap服务，云端即可通过ADNS智能域名解析和DCDN加速，将设备数据分发到就近地域并下发设备接入点域名。

4.3.3. 全球连接加速



依托于云原生架构优势、自建接入层技术优势，阿里云物联网平台在网络层、连接层都进行了大量连接性能优化，在连接并发、建连速度、网络RT及稳定性、数据传输速度等多项关键指标上都进行针对性优化；主要通过以下关键技术解决：

➤ 无限扩展的架构

通过智能DNS和SLB的组合应用解决网络扩展性，应用层使用SLB与ECS，解决接入层的扩展性，并通过单元化架构保障了线性弹性扩展能力，满足数亿级设备的接入要求。

➤ 稳定的网络质量

充分利用云原生优势，基于阿里云BGP网络，网络延迟能达到同城10ms。通过智能DNS的就近接入、就近IP的选址、基于全球加速和DCDN的加速服务进一步提升网络效率。

➤ 极致的应用性能

在应用侧，采用多进程架构和全异步化的通信方式极大提升应用性能，满足百万每秒的建连并发。

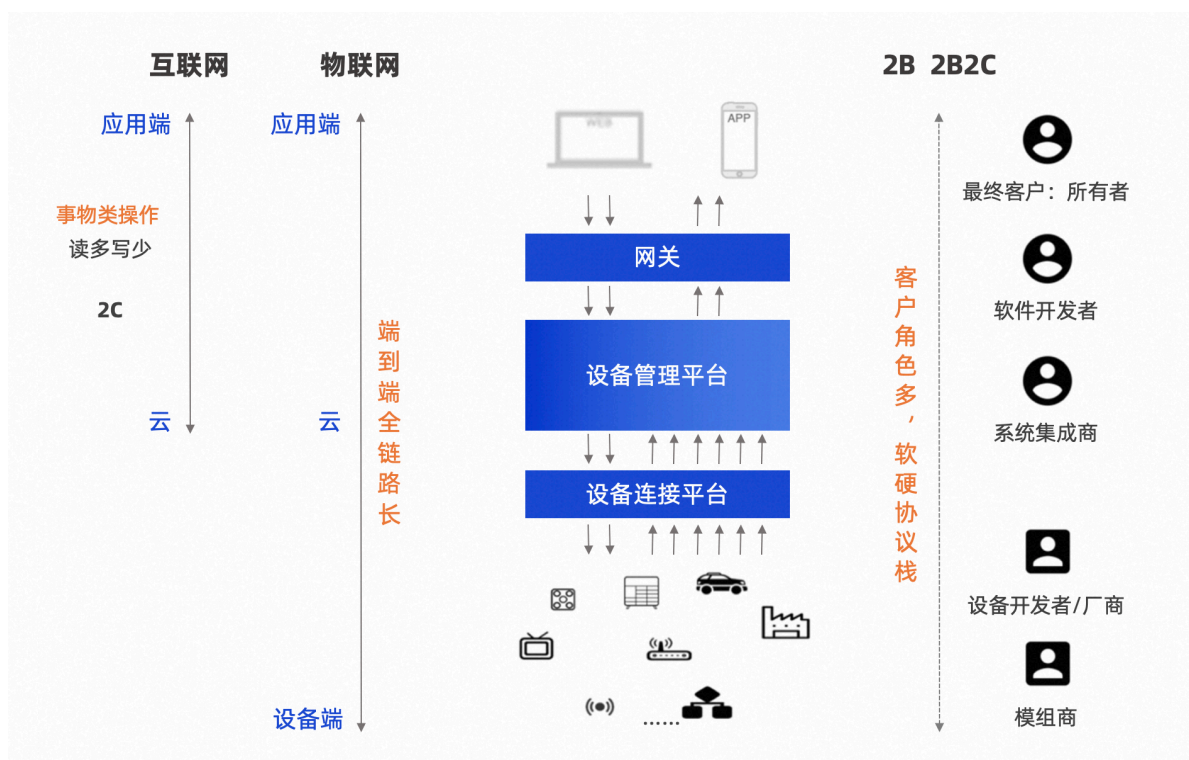
技术	说明
就近接入	通过全球设备分发能力和全球8大地域部署来支持设备的就近接入，做到全球设备接入的网络平均RT在百毫秒内。
连接并发数	通过DNS + NLB的结合，并发连接数可无限扩展，满足用户亿级设备的接入。
网络RT和稳定性	基于阿里云的BGP、全球加速、CDN等网络技术，提升IoT平台的连接质量，网络延迟能达到同城10ms。
数据传输速度	单设备支持多连接，每个连接1 MB/s，单设备可支持100 MB/s的上下行速度。
建连并发	平台集群支持百万的设备并发建连。

05 高可用

5.1. 重要性

物联网平台的设备连接、管理和运维作为PaaS能力提供给客户，客户所属行业场景多，会被用于城市交通、工业制造、商业零售等各种场景，若出故障，可能会导致公交瘫痪、工厂停产、售货柜无法出货等各种情况，对社会民生有非常大的影响，稳定性要求要远高于一般的互联网场景。

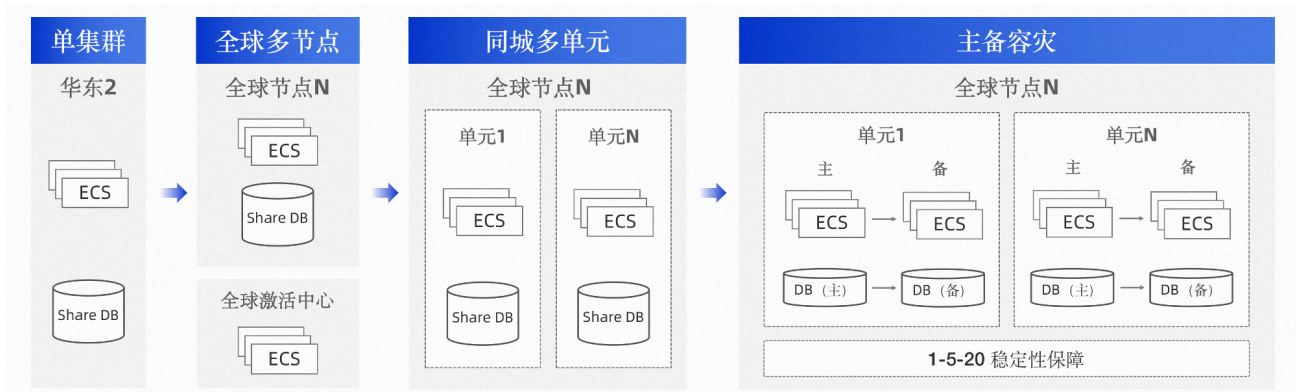
5.2. 挑战性



- 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。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。
- 2) 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。
- 3) 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。
- 4) 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，并且要充分考虑到设备程序质量从而规避云端风暴。

5.3 高可用演进路径

阿里云物联网平台这些年在稳定性建设上持续升级，支撑海量设备已稳定运行多年。主要从以下三方面持续构建高可用能力：



➤ 不断升级高可用架构

- 阿里云高可用架构从单集群架构逐步升级扩展成全球多节点架构、同城多单元架构、主备容灾架构，也即将支持全球容灾。
- SLA达到99.99%。

➤ 全链路保障设备连接稳定性

- 海量长连接稳定可靠，具备发布不断连、优雅下线、防连接风暴、长连接动态负载均衡能力。
- 海量高并发消息可靠推送，实时消息优先、消息回放、QoS多策略保障。

➤ 全面建设1-5-20稳定性保障体系

- 达到故障秒级监测、快速预案、故障恢复全套运维体系和规范。
- 具备一定的故障诊断、故障热迁移能力。

5.4. 单元化架构

物联网平台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。

5.4.1. 技术挑战

➤ toB业务特性带来的技术挑战

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。

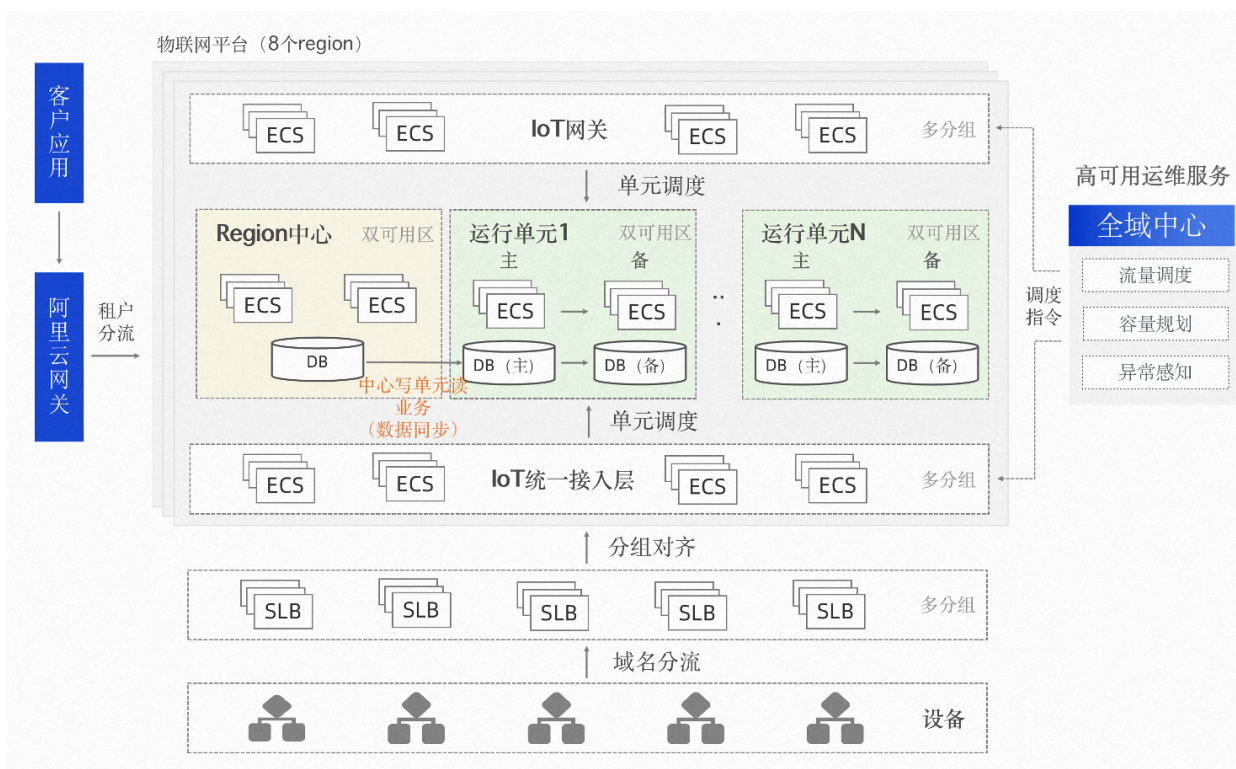
➤ 物联网特性带来的技术挑战

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，同时容灾切换也需要保障端到端全链路流量一致性。

➤ 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。

5.4.2. 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架构拆分为多个运行单元、Region中心，运行单元可线性弹性扩展。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，运行单元同时支持主备容灾能力。

物联网平台为全面升级单元化架构，具备多Available Zone主备容灾能力、单元故障快速感知和摘除能力，核心技术能力主要包括以下几点：

➤ 单元隔离

将流量拆分为多个单元，爆炸半径控制30%以内；单元可线性弹性扩展。

➤ 主备容灾

单元故障可快速Failover；主备双可用区，避免机房级故障。

➤ 流量调度

南北流量统一调度，高可用实例实现百万设备分钟级热迁移；通过动态均衡算法，保障连接和流量的均衡性。

➤ 数据同步

PB级数据强一致同步保障；解决数据冲突、双向同步死循环问题。

➤ 全链路灰度、蓝绿发布能力

可控制变更范围，降低变更对线上客户的影响；从设备连接到应用API访问全链路可灰度。

5.4.3. 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换速度。
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	运行单元主备双副本互为Failover，运行时支持数据同步，并做到严格的数据一致性保障，同时Failover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Region中心和运行单元的双向同步，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。

5.5. 稳定性体系建设

稳定性建设是一整套体系化工程，涉及到架构、机制、稳定性意识、配套运维体系等。阿里云物联网平台围绕降低故障发生概率、减少故障影响范围、提升故障恢复速度三方面对稳定性体系进行建设。

5.5.1. 技术挑战

由于稳定性保障涉及到架构、机制、人员意识、配套运维体系等方面，1-5-20稳定性体系建设就非常复杂，既要做到不出现超大规模故障，又要做到小故障快速发现和恢复。核心挑战点在于：

- 1) 全平台依赖机器、云产品、业务应用都需要全面监控，在漏报率为零的同时还需要尽可能降低噪音。
- 2) 容量、流量的异常巡检，如何避免人工巡检带来的不规范、标准不统一、遗漏问题，具有非常大的挑战。
- 3) 通常随着业务发展、技术构架、代码、监控不断变化，很容易出现稳定性运动式治理，如何避免稳定性治理出现衰退，始终维持稳定性质量的高水位，是做稳定性体系建设最关键的问题。
- 4) 自动化诊断工具，需要围绕设备连接、消息上报、数据存储全业务链路，结合网络、容器等特性提供有效率的诊断工具。

5.5.2. 核心思路



通过事前的架构优化、容量保障、演练保障、变更管控、机制建设；事中的应急快反、容灾保障；事后的复盘沉淀、组织保障，全面提升平台稳定性水位。

5.5.3. 核心技术点

技术	说明
监控全覆盖	全链路的监控覆盖与异常跟进、反馈、归档机制，做到物联网平台核心链路的100%覆盖。全平台依赖机器、云产品、业务应用都进行全面监控，不断调优，做到了漏报率和误报率的较好平衡。
自动化巡检	容量、流量的异常巡检，通过对平台业务、系统异常进行建模和大数据分析，形成每天巡检报告，针对异常点生成自动化任务进行跟进，确保平台容量、流量异常可以快速检测，并进行快速响应。
丰富的运维工具	从稳定性变更工具，到稳定性应急工具、流量治理、诊断等一系列异常诊断工具，帮助快速发现、定位线上问题。
稳定性数字化监测	随着业务发展、技术构架、代码、监控不断变化，很容易出现稳定性运动式治理，要避免稳定性治理出现衰退，达到稳定性高水位常态化，需要做好1-5-20数字化监测，监控、告警、诊断介入、响应率、完成率全面的数字化监测，并定期进行异常分析。



阿里云物联网平台：更快，更稳，更安全

主编：熊益群、王明、张宇、洪飞扬

编委：张程、刘思谦、陈海滨、王霏、彭垚森、黄玮伟、
陈鹤中、葛成、陈曦、莫重、赵佳委、王麦棋

监制：王进、常司晨

设计统筹：仲祐民

市场推广：谢赛微

出品团队：阿里云物联网平台团队

 **阿里云**