阿里云

游戏行业
云上技术服务方案和实践

作者 胡海峰、朱晓健、孙鸿杰、张天慧、徐思婕、梁羿、张瑞、王超、林万境

团队 阿里云-公共云 技术服务部

序言 PREAMBLE

在数字时代的浪潮中，游戏早已超越了单纯的娱乐方式，成为了一个融合技术、艺术与人文的庞大生态系统。从早期的像素游戏到如今的3A大作，游戏产业的每一次飞跃都离不开技术的革新与架构的突破。然而，随着玩家对沉浸感、流畅度和安全性的要求日益提高，游戏开发与运维的复杂性也呈指数级增长。如何构建高效稳定的架构？如何保障千万级玩家的流畅体验？如何在数据洪流中挖掘玩家行为的深层价值？这些问题不仅困扰着开发者，更成为游戏行业持续发展的关键挑战。

本书正是为解答这些问题而生。它不仅是一本技术手册，更是一份实践指南，汇集了游戏开发与运维中的核心领域——从底层架构设计到网络优化，从安全防护到数据驱动的决策支持，再到AI技术对美术创作的革新。书中内容基于作者多年从业经验，结合了大量真实案例与行业前沿技术，旨在为从业者提供一套系统化的解决方案，帮助读者在技术迷宫中找到清晰的路径。

本书的结构设计紧扣游戏开发的全生命周期：从构建虚拟世界的基石——游戏架构，到保障玩家体验的神经网络；从抵御安全威胁的坚固壁垒，到加速内容分发的极速通道；从数据库的灾备恢复艺术，到大数据中挖掘的玩家心声；最终，我们探索了AI技术如何重塑游戏美术的创作范式。每一章节都如同一把钥匙，为读者打开一个技术领域的大门，而将这些钥匙串联起来，便能构建起一座通向成功游戏项目的桥梁。

本书的独特之处在于其实践导向。无论是网络加速的“跨地域隧道”方案，还是数据库恢复的实操演练；无论是CDN高可用架构的搭建，还是AIGC在美术设计中的伦理探讨，均以真实场景为蓝本，提供可复用的策略与工具。我们相信，技术的价值不仅在于理论的完备，更在于其落地的可行性。

希望本书能成为游戏开发者、架构师、运维工程师以及所有对游戏技术感兴趣的读者的良师益友。愿你在阅读中收获启发，在实践中突破边界，共同推动游戏行业向着更广阔的可能性进发。

前言 PREFACE

本书聚焦游戏开发与运维中的核心技术领域，系统性地梳理了从架构设计到安全防护、从数据管理到美术创新的全链条知识体系。以下是对本书内容的简要说明：

第一章：游戏架构——虚拟世界的坚实脊梁

本章从游戏分类入手，深入解析不同类型游戏（如MMORPG、SLG、MOBA等）的系统架构、技术架构与部署架构。通过对比不同游戏类型的底层设计逻辑，揭示如何根据玩法需求选择最优架构方案。无论是服务器集群的分布策略，还是客户端与后端的交互模式，都将为读者提供架构设计的全局视角。

第二章：游戏网络——数字世界的神经与血脉

网络质量直接决定玩家体验的下限。本章探讨如何通过合理的网络拓扑设计、加速技术（如全球加速GA、Anycast EIP）和智能运维工具（如Terraform自动化部署），构建低延迟、高可用的网络环境。同时，结合边缘计算与Serverless架构的未来趋势，为网络优化提供前瞻性思考。

第三章：游戏安全——守护虚拟王国的坚固壁垒

安全是游戏长线运营的生命线。本章剖析不同网络安全架构的优劣，提出高冗余防护、全链路监控与应急预案的“四维防御体系”。通过实战案例，展示如何在攻防演练中筑牢安全防线，确保游戏在流量高峰与DDoS攻击面前坚如磐石。

第四章：游戏下载——加速玩家体验的极速通道

下载环节是玩家接触游戏的第一印象。本章以CDN与OSS为核心，详解高可用架构的搭建方法，涵盖缓存优化、安全防护与监控告警策略。通过资源预热、限流预案等实操技巧，帮助开发者在活动期间从容应对流量洪峰。

第五章：游戏数据库——瞬息恢复的艺术

数据是游戏的核心资产。本章聚焦数据库的灾备与恢复，从误操作导致的数据丢失到跨地域容灾方案，通过实例演练与高频问题解答，传授如何将恢复时间缩短至分钟级。结合云原生工具，为数据安全提供“双保险”。

第六章：游戏大数据——探索玩家心声的数据海洋

从广告投放到社区运营，数据驱动决策已成为行业共识。本章通过两个游戏公司案例，解析如何构建实时湖仓架构，利用Flink等工具挖掘玩家行为数据。同时，探讨大数据在精准营销与内容优化中的实战价值。

第七章：游戏美术——构建幻想世界的画笔与色彩

技术的终极目标是服务于艺术。本章跳出传统技术范畴，探讨AI生成内容（AIGC）对游戏美术的颠覆性影响。从角色设计到场景构建，从版权争议到创意辅助，理性分析AI的潜力与局限，为美术团队提供拥抱新技术的指南。

第八章：游戏内容审核——智能守护虚拟世界的多元表达

内容审核保障玩家自由表达的同时，维护健康、安全、合规的虚拟环境。本章讨论游戏内容审核背景、场景、痛点，探讨了内容审核的方法和技术以及相应的技术调整。通过M游戏公司UGC场景的安全审核方案介绍内容审核的实际落地场景。

第九章：游戏行业的昨天、今天与明天

本章将从“昨天、今天、明天”三个维度，回望游戏行业的技术演进之路，梳理当前云上实践的核心成果，并展望未来十年可能重塑行业的关键技术趋势。

本书特色

- 实战导向：每章均包含真实案例与配置指南，如“东南亚回国加速方案”“数据库恢复演练步骤”。
- 技术深度：覆盖从基础设施到业务逻辑的全栈知识，兼顾架构设计与代码级细节。
- 前瞻视野：讨论边缘计算、Serverless、AIGC等前沿技术对游戏行业的变革作用。

无论是刚入行的游戏开发者，还是经验丰富的架构师，本书都能提供新的视角与实用工具。愿你带着本书的知识，构建更稳定、更安全、更具创意的游戏世界。

致谢

本书的成形离不开众多行业同仁的分享与支持。期待与读者在技术之路上共同成长。书中若有疏漏，欢迎反馈指正。

让我们以技术为笔，以代码为墨，共同书写游戏行业的下一个传奇。

目录 PREFACE

第一章 游戏架构：撑起虚拟世界的坚实脊梁

在电子游戏发展的早期阶段，游戏架构的设计主要围绕着本地硬件资源展开。无论是简单的2D平台跳跃游戏还是复杂的3D开放世界冒险，开发者们往往需要面对有限的计算能力和存储空间。这些限制不仅影响了游戏的表现力和交互深度，还对游戏开发周期、成本以及最终用户体验造成了显著的影响。

随着时间的发展和技术的进步，尤其是互联网的普及和个人电脑性能的大幅提升，网络游戏逐渐兴起。然而，传统的游戏服务器架构依然面临着诸多挑战——从高昂的硬件维护成本到难以扩展的基础设施，再到应对全球范围内玩家并发访问时可能出现的服务中断问题。这些问题迫使游戏开发者不断寻求更高效、更灵活的解决方案。

进入云计算时代，这一切发生了根本性的变化。以阿里云为代表的云计算平台提供了前所未有的强大支持，使得游戏开发者能够构建出更加动态、可扩展且高度可靠的游戏环境。通过利用阿里云提供的丰富服务，如弹性计算、分布式数据库、智能网络优化以及全面的安全防护机制，开发者现在可以专注于创造卓越的游戏体验，而无需担心底层基础设施的复杂管理和维护。

游戏的类型非常多，不同的游戏分类维度，对于游戏架构设计的考虑有差异，例如端游和手游对底层资源要求不同、MOBA类型和SLG类型对网络延迟要求不同，因此我们基于不同的分类维度，去思考架构设计和对应资源的差异点。

1.1 游戏分类

1.1.1 游戏运行平台分类

根据游戏运行平台维度划分游戏类型，可以有以下几种：

◆ **PC 游戏（个人电脑游戏，简称端游）**

- 高性能要求：由于PC硬件配置可高度定制化，许多PC游戏提供了高画质、复杂的物理引擎以及精细的画面效果。
- 多样性与深度：支持多种输入设备（如键盘、鼠标、手柄等），适合各种类型的游戏玩家。尤其在策略类、模拟类游戏中表现突出。
- 社区与 Mod 支持：很多 PC 游戏拥有活跃的玩家社区，并且支持用户自定义内容(Mod)，这极大地延长了游戏寿命

◆ **主机游戏（PlayStation, Xbox, Nintendo Switch 等）**

- 优化体验：专门为特定硬件设计，确保最佳性能和稳定性。通常提供流畅的游戏体验，无需担心兼容性问题。
- 家庭娱乐中心：主机往往被视为家庭娱乐系统的一部分，除了玩游戏外，还可以播放电

影、音乐等多媒体内容。

- 社交互动：通过网络服务（如 PlayStation Network 或 Xbox Live），玩家可以轻松地与其他玩家进行联机对战或合作。

◆ 移动游戏（智能手机和平板电脑,简称手游）

- 便捷性与普及度：几乎人人都有智能手机，使得移动游戏成为最容易接触的游戏形式之一。随时随地都可以玩，碎片化时间利用效率高。
- 轻量级设计：为了适应移动设备的硬件限制，这类游戏通常设计得较为简单，操作也相对直观，易于上手但具有深度挑战性。
- 免费增值模式：很多移动游戏采用免费下载+内购的形式，鼓励玩家通过小额支付购买虚拟物品或解锁新内容。

◆ 网页游戏

- 无需安装：直接在浏览器中运行，不需要额外安装软件，降低了进入门槛。
- 跨平台能力：理论上可以在任何能上网的设备上运行，包括 PC、平板甚至部分智能电视。
- 社交元素强：许多网页游戏强调社交互动，例如农场类游戏中的好友互助功能，或是 MMORPG 中的团队作战。

◆ 云游戏

- 无需高端硬件：所有计算都在云端完成，玩家只需具备基本的流媒体接收设备即可享受高质量游戏体验。
- 即时访问：无需下载或安装，点击即玩，大大缩短了开始游戏的时间。
- 多设备兼容性：可以在不同类型的设备间无缝切换，无论是手机、平板还是电视，都能获得一致的游戏体验。

◆ VR/AR游戏（虚拟现实/增强现 实游戏）

- 沉浸式体验：借助专用头戴显示器或其他穿戴式设备，为玩家带来前所未有的身临其境之感。
- 交互方式创新：允许玩家通过手势、头部动作甚至是语音来控制游戏角色或环境，增加了游戏的真实感和趣味性。
- 技术挑战：尽管前景广阔，但目前仍面临诸如成本高昂、内容不足及晕动症等问题。

每种类型的游戏都有其独特之处，选择哪种类型取决于玩家的兴趣偏好、可用资源以及期望的游戏体验。随着技术的发展，不同类型之间的界限也在逐渐模糊，越来越多的游戏开始尝试融合多种元素，以提供更加丰富多元的游戏体验。

1.1.2 游戏玩法类型

根据游戏机制和玩家互动方式区分，游戏类型具有独特的玩法逻辑、互动模式、游戏特点，并且覆盖了不同的发行端，包括手游、端游、页游等。以下列举最主要也是最常见的游戏玩法类型：

◆ **MMO (Massively Multiplayer Online) 大型多人在线游戏**

- 玩法逻辑：在一个共享的虚拟世界中，玩家可以自由探索、完成任务、与其他玩家交互。
- 互动模式：高度社交化，支持大规模在线玩家互动。
- 游戏特点：开放世界，持续更新内容，强调社区和团队合作。
- 发行端：常见于端游，也有手游 MMO。
- 游戏示例：《魔兽世界》、《最终幻想 XIV》。
- 售卖模式：多采用免费+内购，或订阅制。

◆ **RPG (Role-Playing Game) 角色扮演游戏**

- 玩法逻辑：玩家扮演一个虚构角色，在游戏世界中成长、探索、完成任务。
- 互动模式：单人游戏为主，但也包含多人合作或对战模式，注重角色发展和故事体验。
- 游戏特点：丰富的故事情节，角色定制和成长系统。
- 发行端：广泛应用于端游、手游、页游。
- 游戏示例：《黑神话悟空》、《真叁国无双》、《巫师 3》、《勇者斗恶龙》系列。
- 售卖模式：免费+内购或一次性购买。

◆ **MMORPG (Massively Multiplayer Online Role-Playing Game) 大型多人在线角色扮演**

- 玩法逻辑：结合了MMO和RPG的特点，大型在线环境中扮演角色，进行角色扮演和社交。
- 互动模式：高度社交，支持大规模PvP和PvE活动，强调团队合作。
- 游戏特点：持续的世界发展，复杂的经济系统，拥有丰富的故事背景、角色定制。
- 发行端：主要为端游，也有手游尝试。
- 游戏示例：《仙剑世界》、《魔兽世界》、《剑网3》。
- 售卖模式：通常为免费+内购或订阅制。

◆ **SLG (Strategy Game) 策略游戏**

- 玩法逻辑：强调策略规划与资源管理，玩家需制定长期计划，如城市建设、军队部署、外交策略等。
- 互动模式：通常包括PvP（玩家对玩家）和PvE（玩家对环境），玩家间可结盟或对抗。
- 游戏特点：慢节奏，重思考，适合喜欢策略规划的玩家。
- 发行端：广泛存在于手游、端游。
- 游戏示例：《万国觉醒》、《部落冲突》、《文明系列》。
- 售卖模式：通常为免费下载+内购，如加速建造、资源包等。

◆ **FPS/TPS (First-Person Shooter/Third-Person Shooter) 第一人称/第三人称射击游戏**

- 玩法逻辑：以射击为核心，玩家通过视角控制角色进行战斗。
- 互动模式：PvP对抗为主，也有单人战役模式(PVE)。
- 游戏特点：快节奏，强调即时反应和战术配合。

- 发行端：广泛存在于端游和手游。
- 游戏示例：《使命召唤》、《绝地求生》（FPS），《战争机器》（TPS）、《穿越火线》、《逆战》、《和平精英》。
- 售卖模式：多为一次性购买或免费+内购皮肤、武器等。

◆ **MOBA (Multiplayer Online Battle Arena) 多人在线战术竞技游戏**

- 玩法逻辑：两队玩家控制各自英雄，在固定地图上进行对战，目标通常是摧毁对方基地。
- 互动模式：高度团队合作，PvP对战。
- 游戏特点：强调英雄技能搭配、团队协作和战术策略。
- 发行端：广泛存在于端游和手游。
- 游戏示例：《英雄联盟》、《王者荣耀》。
- 售卖模式：免费下载+内购皮肤、英雄等。

◆ **AVG (Adventure Game) 冒险游戏**

- 玩法逻辑：通常围绕着一个故事展开，玩家通过解谜、探索环境、与NPC对话等方式推进剧情。游戏强调叙事和角色发展。
- 互动模式：互动主要集中在剧情选择、物品收集和使用上，玩家的选择会影响故事走向和结局。
- 游戏特点：丰富的剧情，强调探索和解谜。
- 发行端：多见于端游和手游。
- 游戏示例：《生化奇兵》、《神秘海域》。
- 售卖模式：可能采用付费下载、订阅制或混合模式，尤其是对于高质量、内容丰富的游戏。

◆ **SIM (Simulation) 模拟游戏**

- 玩法逻辑：SIM游戏模拟现实或虚构情境，如城市建造、经营农场、生活模拟等，玩家需管理资源、制定策略以达成游戏目标。
- 互动模式：侧重于策略规划和资源管理，玩家与游戏系统的互动更多体现在决策制定上，如经济调控、设施布局等。
- 游戏特点：高度还原现实，强调管理和策略。
- 发行端：广泛存在，尤其在端游和手游。
- 游戏示例：《心动小镇》、《模拟人生》、《模拟城市》。
- 售卖模式：可能采用付费下载、订阅制或混合模式，尤其是对于高质量、内容丰富的游戏。

◆ **沙盒游戏**

- 玩法逻辑：提供一个开放的世界，玩家拥有极高的自由度，可以自由探索、创造、改变游戏环境，没有固定的目标或线性剧情。
- 互动模式：强调玩家与游戏世界的互动，包括建造、破坏、社交互动等，玩家行为对游戏世界产生直接影响。
- 游戏特点：极高自由度，无固定目标。
- 发行端：多见于端游、手游。
- 游戏示例：《迷你世界》、《我的世界》、《GTA V》开放世界模式。

- 售卖模式：免费+内购，利用游戏内购买皮肤等来盈利，通过持续的内容创造和社区互动吸引玩家长期投入。

每种游戏类型都有其独特的魅力和受众群体，跨平台发行也为玩家提供了更加丰富多元的游戏体验。

1.2 游戏架构概念

游戏架构是一个综合概念，构建和组织游戏各个组件的方式，确保这些组件能够高效协作，提供流畅、稳定且可扩展的游戏体验，良好的游戏架构不仅有助于提高开发效率，还能增强游戏的性能和可维护性。游戏架构不仅仅局限于单一层面，而是涵盖了游戏设计与技术实施的多个维度。具体来说，**游戏架构**包括了**游戏系统架构、游戏技术架构**和**游戏部署架构**：

1.2.1 游戏系统架构

系统架构跟游戏内容和游戏设定强相关，通常在游戏开发的早期阶段，即概念设计与预制作阶段就需要初步确定。系统架构是所有游戏开发工作的基础，指导着技术选型、团队分工、资源分配等关键决策。但是系统架构并非一成不变，随着开发过程中的反馈和技术演进，架构可能会经历迭代和优化。特别是在技术测试期间，通过实际运行数据和玩家反馈，可以对系统架构进行必要的调整，以确保最终产品的稳定性和用户体验。

游戏系统架构的设计受到多种因素的影响，这些因素共同决定了架构的形态和复杂度。以下是决定游戏系统架构的关键因素：

1. 游戏类型与玩法：不同的游戏类型（如MMORPG、MOBA、FPS、休闲游戏等）对系统架构的需求差异显著，例如，MMORPG需要强大的服务器集群来处理大量玩家的实时交互和复杂的游戏世界状态，而休闲游戏可能更侧重于客户端的优化和快速加载；

2. 目标用户群体：用户的地理位置、设备类型（PC、移动）、网络条件等，影响着游戏的网络架构设计、客户端性能优化以及多语言支持等；

3. 并发量与扩展性：预期的同时在线玩家数、高峰时段的流量处理能力，要求架构具备良好的水平扩展性和负载均衡机制，以应对突发流量和长期增长；

4. 安全性：保护用户数据、防止作弊、确保交易安全等，需要在架构中集成安全措施，如数据加密、DDoS防护、反欺诈系统等；

5. 经济系统与虚拟商品管理：对于含有内购或虚拟货币的游戏，系统架构需支持复杂的经济模型、交易记录、库存管理等；

6. 社交与社区功能：游戏内的聊天、好友、公会、排行榜等功能，要求架构支持高效的即时通讯和社交网络设计；

7. 内容更新与版本管理：频繁的内容更新和版本迭代，要求架构支持快速部署、热更新、版本回滚等机制；

8. 技术栈与开发工具：团队的技术专长、第三方服务集成（如云服务、数据库、中间

件）、开发工具的选择，也会影响架构设计；

9. 成本与运维：预算限制、运维效率、基础设施成本等因素，促使架构设计时考虑成本效益，比如采用云服务以减少初期投资和运维负担；

10. 合规性与地域要求：不同国家和地区的法律法规（如GDPR、儿童在线隐私保护）对数据存储、处理有特定要求，影响架构的合规设计；

综上所述，游戏系统架构关注游戏逻辑和内容的设计，包括游戏世界的设计、角色系统、物品系统、交互系统（如PvP、PvE）、经济系统、成就与奖励机制等。它定义了游戏的玩法、规则以及玩家与游戏世界的互动方式，是一个综合考量技术、市场、运营、成本等多方面因素的复杂决策过程，需要在项目初期就进行深入分析和规划。

1.2.2 游戏技术架构

游戏技术架构涵盖了多个关键领域，包括服务端架构（即游戏后台服务）、客户端架构、数据处理架构以及网络通信架构等核心模块。此外，还涉及安全防护、数据分析与监控、运维自动化等方面。下面我们将详细介绍这些核心模块。

◆ **服务器架构：游戏逻辑的中枢**

服务器架构是游戏技术栈的基础，主要负责处理游戏逻辑、状态同步和数据存储等功能。具体来说：

- **游戏逻辑服务器：**承担着执行游戏规则、更新状态及触发事件等任务，确保游戏世界的正常运作。例如，在MMO游戏中，场景服务器管理特定地图内的玩家交互逻辑。它分为两部分：
  - **场景服务器：**包含游戏的核心逻辑，如规则执行、事件触发等，对CPU处理能力和网络包转发能力有较高要求。
  - **游戏世界管理：**根据地图或场景划分逻辑服务器，当单个区域玩家过多时，通过分线或多服务器方式扩展。
- **网关服务器：**作为客户端与游戏服务器之间的桥梁，负责网络数据包的转发，处理登录验证、负载均衡等工作，是网络流量的关键节点，对网络吞吐量要求高。
- **数据中心服务器：**用于缓存玩家数据，提高访问速度，并异步处理数据持久化，保证数据的高可用性和一致性。
- **分布式部署：**为了满足高并发和全球覆盖的需求，游戏服务器通常采用分布式架构，通过多区域部署和数据同步机制优化玩家体验和数据一致性。

◆ **客户端架构：玩家体验的直接体现**

客户端架构决定了玩家的交互体验和游戏性能，主要包括开发框架、渲染引擎、输入处理和用户界面设计等方面：

- **开发框架：**包括游戏引擎、UI框架和脚本语言。
  - **游戏引擎：**如Unity、Unreal Engine等，提供了图形渲染、物理模拟、脚本编写等功能，简化了游戏开发流程。Unity因其跨平台能力广受青睐，而Unreal Engine则以其高级视觉效果闻名。
  - **UI框架：**如Unity的UGUI、Unreal的UMG，用于设计游戏内用户界面，确保良好用户体验。

- **脚本语言：**如C#（Unity）、Blueprints/UnrealScript（Unreal Engine）、Lua（Cocos2d-x），用于编写游戏逻辑和交互控制。

- **渲染引擎：**将游戏中的场景、角色、特效转化为可视图像，涉及图形渲染和优化技术。
  - **图形渲染：**负责将3D模型、纹理、光照等转换为屏幕上的图像，支持PBR、全局光照等高级渲染技术。
  - **优化技术：**包括LOD、遮挡剔除、动态加载等，平衡画质与性能。
- **输入处理：**处理用户操作的即时反馈，如角色移动、攻击等，包括输入系统和响应性化。
- **用户界面：**展示游戏画面并接收用户输入，涉及交互设计、视觉设计和适配性，需简洁直观且符合玩家习惯。

◆ **数据处理架构：高效的数据管理和访问**

数据处理架构专注于如何存储和访问游戏数据，如玩家信息、游戏进度、物品库存等：

- **数据库选择：**使用云数据库RDS等存储玩家数据，常采用关系型数据库（如MySQL）和非关系型数据库（如MongoDB）结合的方式。
- **数据缓存：**利用Redis等技术提高数据读取速度，减轻数据库压力。
- **数据异步处理与批处理：**对于非实时性要求的操作，采用异步处理和批量处理策略，提升系统吞吐量。
- **监控数据采集处理：**收集业务逻辑日志、性能监控指标等数据，支持决策优化。

◆ **网络通信架构：保障高效安全的数据传输**

网络通信架构确保游戏数据在客户端与服务器之间高效、安全地传输：

- **协议选择：**根据游戏类型选择合适的网络通信协议，如TCP适用于数据可靠性高的场景，UDP适合实时性要求高的应用。
- **网络优化：**采用预测算法、客户端预测与服务器校验、数据压缩等技术减少延迟。
- **全球部署与CDN：**通过CDN加速静态资源分发，为全球玩家提供低延迟接入。

此外，还包括安全防护、数据分析与监控、运维自动化等模块，共同构建一个稳定、高效的游戏服务平台。安全防护模块实施权限控制、数据加密等措施；数据分析与监控模块收集运行数据进行性能监控和玩家行为分析；运维自动化模块则致力于构建自动部署、故障恢复等体系，确保游戏服务的稳定运行。

1.2.3 游戏部署架构

良好的技术架构可以确保游戏在高并发时稳定运行，而合理的部署架构则能有效降低成本并提升服务的全球可达性。游戏的部署架构不能简单的归为“集中部署”or“分散部署”，一款完整游戏的架构包括游戏服、平台服、数据服，其中数据服还包括在线和离线计算两部分。此外还有较独立于游戏自身逻辑的数据更新服（常见服务如OSS、CDN等）、数据运营服（常见服务如ADB、EMR、Quickbi等）、游戏运维平台（常见服务如云监控、SLS、RAM、短信、邮件等）。

- 平台服(逻辑服)：账号、充值、聊天、社区等
- 游戏服：对战/房间、游戏逻辑服
- 小服模式（分区分服）：一台服务器即一组Game Server，可承载很多玩家，甚至自带登

陆、数据库缓存、排行等

- 大服模式（全区全服）：按照游戏服务类型拆分，同类服务采用集群模式，可扩展，如：大厅、战斗、地图等，通常有网关
- 集群：同一类服务采用多台物理/云主机部署，无状态居多，可能配备集群自身专用的DB和缓存，可弹性扩缩容

游戏部署架构的设计需综合考虑游戏类型、玩家分布、网络时延要求、数据合规性以及成本效益等因素。以下是一个基于阿里云产品的游戏部署架构设计思路，涵盖平台逻辑服、数据服、游戏服、以及可能的转发服（网关）的部署策略。部署架构的关键考虑因素包括：

于阿里云产品的游戏部署架构设计思路，涵盖平台逻辑服、数据服、游戏服、以及可能的转发服（网关）的部署策略。部署架构的关键考虑因素包括：

◆ **网络延迟要求**

- **低时延要求（>200ms） - 平台服和游戏服集中部署**
  - **适用游戏类型：**大部分SLG（策略类游戏）、休闲类游戏、棋牌游戏（非实时对战）、养成类RPG等。
  - **部署策略：**将游戏服、平台服（核心逻辑+数据）集中在某个中心Region，如美东、法兰克福或香港，利用阿里云GA全球加速确保大部分玩家的接入时延在可接受范围内保障游戏体验。对于国内运维，可以通过GA加速或CEN（云企业网）构建境内外游戏运营内网，保证运维访问质量。
- **中等时延要求（100-200ms） - 游戏服分散部署，平台服可集中可分散**
  - **适用游戏类型：**RPG（角色扮演类游戏）、部分实时对战游戏。
  - **部署策略：**游戏服需要根据玩家的主要分布区域分散部署，确保玩家能够就近接入，减少网络延迟。平台服可以集中部署也可以根据需要分散部署。例如，RPG游戏服分散部署在亚洲、北美和欧洲的主要数据中心，同时利用阿里云的全球网络和数据同步机制保证数据一致性。敏感数据需落本地，遵守相应国家的数据监管要求。
- **高时延要求（<100ms） - 分布式部署**
  - **适用游戏类型：**MOBA类游戏、竞技类、射击类游戏等对实时交互要求极高的游戏。
  - **部署策略：**游戏逻辑服需要在全球多个区域分散部署，确保玩家间的交互延迟控制在最低水平。这种架构下，每个区域的游戏逻辑服负责处理该区域玩家的请求，通过高效的网络同步机制保持数据一致性。同时，利用GA全球加速确保玩家的接入速度，以及数据同步解决跨区域玩家互动延迟问题。

表1

	游戏服集中	游戏服分散
平台服集中 （数据和逻辑均集中）	适合延迟要求较低的游戏 游戏类型：单机、益智、一部分MMO、大部分SLG、卡牌、挂机	适合延迟要求中等的游戏 游戏类型：MMO、SLG、ARPG、卡牌等
平台服半集中 （数据集中、逻辑分散）	--	适合延迟要求非常高的游戏 游戏类型：FPS、MOBA
平台服分散 （数据和逻辑均分散）	--	通常用于有分区域发行要求的游戏， 或延迟要求非常高的游戏 数据同步难度大，跨大区游戏交互受限

◆ **数据合规要求**

数据服部署架构，最关键的考虑因素就是合规。玩家数据和个人信息需遵循当地法律法规，如越南要求数据落本地。因此，在玩家密集区域选择大型数据中心部署服务集群及数据库（如PolarDB/RDS）。同时，利用阿里云的全球网络CEN确保数据同步的高效性和安全性。

- **敏感数据本地化：**根据各国数据保护法规（如欧盟的GDPR、越南的数据本地化要求），在涉及玩家个人信息处理时，需在该国或邻近地区部署数据库集群，确保数据存储符合当地法律要求。
- **加密传输与存储：**无论数据集中还是分布部署，均应采用SSL/TLS加密通信，以及数据库和服务端的数据加密存储，确保数据在传输和存储状态下的安全。

◆ **用户区域分布**

- **全球加速网络：**使用阿里云GA（全球加速）服务，确保玩家无论身处何地都能通过最近的POP点接入，通过阿里云全球骨干网高效回源至游戏服务器，解决跨国网络延迟和丢包问题。
- **区域优化部署：**根据玩家分布密集度选择数据中心，如北美优先美东，欧洲选法兰克福，日韩优先东京，东南亚以新加坡为中心，确保大多数玩家能接入最近的服务器。
- **转发服（网关服）：**转发服作为玩家与游戏服之间的桥梁，负责负载均衡、安全防护和协议转换等功能,应部署在每个主要区域，确保玩家请求能快速转发至最近的游戏服。

◆ **扩展性和弹性**

随着玩家数量的波动，游戏服务器需要能够快速扩展以应对高峰流量，同时在低峰期自动缩减资源以节省成本。本身做架构设计时考虑扩展性，尤其是分布式架构的扩展性，确保系统能够高效地应对增长的业务需求、用户量和数据量，同时保持或提升性能和服务质量。以下是几个关键原则和策略，用于确保架构具备良好的扩展性：

- **可扩展原则**

纵向扩展（Scale Up）：增加单个资源的处理能力，例如选择更高配置的云服务器，提升负载均衡器规格。

横向扩展（Scale Out）：增加资源的数量来分散负载，例如增加更多的服务器、数据库实例或负载均衡，利用云平台弹性伸缩，根据预设的规则自动增加或减少计算资源，以应对流量波动。

- **分布式设计**

微服务架构：将大型应用拆分为一组小型、独立的服务，每个服务负责特定的功能。这不仅提高了系统的可维护性和可扩展性，还使得各个服务可以根据需要独立扩展。

数据分片与分区：对于数据库层，可以采用数据分片（Sharding）或分区（Partitioning）策略，将数据分布在不同的数据库节点上，以支持更高的读写吞吐量和存储容量。例如，PolarDB分布式版提供的水平拆分和垂直拆分能力，能有效应对大规模数据处理需求。

服务与数据的多区域分布：在多地域部署服务和数据副本，可以减少延迟，提高可用性，并且通过负载均衡在不同区域间分配请求，实现区域扩展。

- **中间件与服务治理**

使用服务网格、API网关、消息队列等中间件来优化服务间的通信、负载均衡和故障隔离，确保系统在扩展时的高可用性和一致性。实施服务治理策略，如服务发现、健康检查、负载均

衡和熔断机制，以动态管理服务实例，确保系统的稳定性和弹性。

- **云原生技术栈**

采用容器化以及云原生服务（如Nacos服务发现与配置管理），天然支持动态扩展和快速部署，有助于构建高度可扩展的系统。

总之，架构的扩展性设计应综合考虑多种策略和技术，确保系统能够在不牺牲性能和稳定性的情况下，高效地应对业务增长带来的挑战。

◆ **灾备高可用**

对于平台服尤其是核心服务，有必要做主备高可用架构部署设计，以确保在主区出现问题时能迅速切换，减少服务中断时间。利用阿里云的多可用区部署能力，实现服务层和数据层的高可用性。

◆ **成本与运维**

分散部署虽然能降低延迟，但会增加运维复杂度和成本，需权衡利弊。成本上用云服务的弹性资源管理，如ECS的自动伸缩组，根据游戏流量动态调整资源，避免高峰期资源不足或低谷期资源浪费。

运维自动化：自动化运维工具和DevOps实践可以提高部署效率，减少人为错误，确保快速迭代和高效运维。建立全面的监控体系，包括性能监控、日志分析、安全监控等，利用阿里云ARMS（应用实时监控服务）和SLS（日志服务）等工具，确保问题能被快速发现并响应。

游戏部署架构的选择应紧密贴合游戏的实时性需求、玩家分布特征以及业务发展策略，充分利用云服务的全球化布局和网络加速能力，确保玩家获得流畅的游戏体验。

1.3 游戏架构设计

每款游戏的玩法逻辑和互动模式，一定程度上决定了云资源产品依赖、性能要求、数据处理分析的时效性要求。以下将列举几种常见类型的游戏架构设计，根据游戏的特点，说明如何选择合适的架构和云产品方案。

MMORPG（大型多人在线角色扮演游戏）、**SLG**（策略游戏）、**MOBA**（多人在线战术竞技游戏）**和放置经营类游戏，在云资源产品依赖、性能要求以及数据处理分析的时效性要求上有一些共性，同时也各有侧重。**

1.云资源产品依赖

- 计算（ECS/ACK）：游戏都需要计算能力来处理游戏逻辑、物理模拟、AI运算等。**特别是MMORPG和MOBA游戏，由于实时交互和复杂的场景渲染，对CPU和GPU的需求更高。**
- 存储（OSS& ESSD）：游戏资源（如地图、模型、音频、视频）需要大量的存储空间。OSS适合存放静态资源，而ESSD云盘则为游戏服务器提供高速的读写能力，确保游戏流畅运行。
- 网络（SLB& CDN）：为了保证全球玩家的低延迟体验，SLB用于负载均衡，CDN用于内容分发，尤其对于全球部署的MMORPG和MOBA游戏更为重要。
- 数据库（RDS/polardb/mongodb）：游戏数据存储，如用户账户、游戏进度、排行榜等，RDS/polardb适用于关系型数据，mongodb一般存储玩家数据。

- 弹性伸缩（Auto Scaling）：对于SLG和放置经营类游戏，玩家在线数量可能随时间波动较大，弹性伸缩能根据实际需求自动调整资源，降低成本。

2.主要性能要求

- MMORPG：强调高并发处理能力、实时交互和图形渲染性能，需要高性能GPU实例和低延迟网络。
- SLG：虽然对实时性要求不如MMORPG，但对策略计算和大数据分析能力有较高要求，需要稳定的计算资源和高效的数据处理能力。
- MOBA：要求极低的延迟和高度的实时交互，对服务器响应速度、网络带宽和计算能力有严格要求。
- 放置经营：相对而言对性能要求较低，但需保证24小时稳定运行，对服务器的稳定性和数据持久性要求较高。

3.数据处理分析时效性要求

- 实时数据分析：对于MMORPG和MOBA，实时数据分析（如玩家行为分析、战斗数据统计）对于即时调整游戏平衡、识别作弊行为至关重要，需要实时计算服务如EMR Flink来处理。
- 运营离线分析：SLG和放置经营类游戏更侧重于长期的用户留存、付费行为分析，虽然时效性要求不如实时战斗数据那么高，但也需要快速的数据处理能力以支持快速决策，如使用MaxCompute进行批量分析。
- 个性化推荐：所有类型的游戏都可能利用用户行为数据进行个性化内容推荐，这要求数据处理具有一定的实时性，以便快速响应用户行为变化。

综上所述，不同类型的游戏在云资源依赖和性能要求上有所差异，但普遍重视计算能力、存储、网络性能以及数据处理的时效性，以确保玩家体验和运营效率。

MMORPG（大型多人在线角色扮演游戏）、SLG（策略游戏）、MOBA（多人在线战术竞技游戏）、放置经营类这几种游戏类型在系统架构、技术架构、部署架构的设计上各有侧重，以适应各自游戏特性和玩家需求。

1.3.1 MMORPG

前面有提到，MMORPG（Massively Multiplayer Online Role-Playing Game，大型多人在线角色扮演游戏）结合了MMO（大型多人在线）和RPG（角色扮演游戏）的特点。这类游戏允许大量玩家在一个持续发展的虚拟世界中共同参与，每个玩家控制一个自定义的角色，进行探索、完成任务、与其他玩家互动以及角色发展。在架构上，MMORPG与普通网络游戏的主要区别在于其规模和复杂性，具体体现在以下几个方面：

服务器架构

MMORPG通常采用分布式服务器架构来处理大量并发玩家。这包括但不限于：

- 区域服务器：将游戏世界分割成多个区域，每个区域由一个或多个服务器管理，以减少单个服务器的压力。
- 登录服务器：处理玩家的登录验证和角色列表显示，然后将玩家分配到相应的游戏世界服务器。
- 数据库服务器：存储玩家数据、游戏物品、地图信息等，通常需要高性能和高可用性设计以应对大量读写操作。

- 聊天服务器：处理游戏内的文字或语音通信，可能独立于游戏逻辑服务器。

网络延迟管理

由于玩家遍布全球，减少网络延迟对于提升游戏体验至关重要。这可能涉及：

- CDN：用于加速静态资源的加载，如游戏客户端文件、图像和音频。
- 全球数据中心部署：在不同地理位置部署服务器，使玩家能够连接到最近的服务器，减少延迟。
- 网络优化：如TCP/UDP协议优化、数据包压缩、预测性移动等，以减少数据传输时间和提高响应速度。

数据同步与一致性

MMORPG需要复杂的同步机制来确保所有玩家看到一致的游戏状态，尤其是在PvP（玩家对玩家）交互和大规模多人事件中。这可能包括：

- 状态同步：定期或根据事件触发，更新玩家位置、状态等信息到所有相关玩家。
- 事务处理：确保交易、战斗结果等关键操作的原子性和一致性。
- 本地缓存与回写机制：减少对中心数据库的依赖，提高响应速度，同时确保数据最终一致性。

可扩展性与容错

- 随着玩家数量的增长，系统需要能够动态扩展资源。此外，高可用性设计确保即使部分组件失败，游戏也能继续运行，这包括负载均衡、故障转移和冗余系统。游戏逻辑服务器可能设计为无状态，便于水平扩展。
- 数据库设计上可能采用分片集群，如云MongoDB分片集群独享型云盘版，以支持高并发读写。

MMORPG的架构设计不仅要满足大量玩家的同时在线需求，还要确保游戏世界的连贯性、玩家间的实时互动，以及在全球范围内的低延迟体验，这些都构成了其架构上的独特之处。

系统架构

强调持续的世界观和角色成长，系统架构需要支持复杂的社交系统、经济系统、大规模的开放世界探索和实时交互。通常采用客户端-服务器架构，服务器端包含多种服务模块，如世界服务器、角色服务器、战斗服务器等，以支持大量并发玩家和复杂的游戏逻辑。

技术架构

由于游戏世界庞大，常采用分布式微服务架构，以支持大规模数据处理和高并发访问。重视数据缓存、数据库分片和负载均衡技术，以应对大量玩家的同时在线和数据交互。

部署架构

由于玩家遍布全球，部署架构倾向于全球分布式，利用多地域部署和全球加速服务来减少延迟。可能还会根据玩家密集区域设置数据中心，以优化用户体验。

1.3.2 SLG

SLG（策略类游戏）强调策略规划和长期发展，可能更侧重于后端数据处理和分析能力。数据库选择上可能会更重视事务支持和数据分析能力，如使用RDS MySQL。服务器架构可能需要支持大量并发的策略计算和存储海量玩家数据。在服务器架构、网络延迟管理、数据同步与

一致性、可扩展性与容错等方面，相比其他类型游戏（如RPG、FPS、MOBA等）有一些特定的设计考虑和区别：

服务器架构

- 集中部署，全球覆盖：由于 SLG 游戏对实时性要求相对较低，通常可以采用集中部署的方式，将游戏服务器和平台服务器（包含核心逻辑和数据）集中在少数几个区域，通过全球加速网络确保玩家体验。
- 平台服集中或半集中：SLG 游戏的平台服务（如账户管理、排行榜等）倾向于集中管理，以简化数据处理和逻辑控制，而游戏逻辑可能根据需要在逻辑上集中或半集中，以适应不同地区的数据合规要求。
- 与其他类型游戏对比，对比 RPG、FPS 等对实时交互要求极高的游戏，SLG 较少采用分布式架构，减少跨区域数据同步的复杂性。

网络延迟管理

- 网络加速优化：虽然 SLG 对延迟容忍度较高，但仍然需要通过全球加速服务（如阿里云 GA）来优化玩家的接入体验，确保全球玩家都能快速连接到游戏服务器。相比于 FPS、MOBA 等游戏对低延迟的严格要求（通常<100ms），SLG 的网络策略更侧重于全局覆盖而非极致的低延迟。

数据同步与一致性

- 敏感数据落本地：考虑到数据合规性，SLG 游戏在特定区域（如越南）可能需要将玩家个人信息存储在当地，但游戏体验仍连接至中心服务器，这要求有高效的数据同步机制。
- 数据同步难度较低：由于 SLG 游戏中的实时交互需求较低，数据一致性要求不如 RPG 或 MOBA 游戏严格，同步策略可以更加宽松。
- 对比 RPG、MOBA 等游戏，这些游戏往往需要高度一致的实时数据同步，以支持即时战斗和交互，SLG 的同步压力较小。

可扩展性与容错

- 集群部署：虽然 SLG 集中部署，但游戏服和平台服依然采用集群模式，便于弹性扩缩容，以应对玩家数量波动。
- 容错设计：SLG 游戏同样重视容错机制，确保单点故障不会影响整体服务，但由于游戏类型特性，容错设计可能更侧重于数据和服务的稳定性，而非实时交互的连续性。相较于需要高度动态调整和即时响应的 FPS、MOBA 游戏，SLG 在可扩展性和容错设计上可能更注重长期稳定性和成本效益。
- 高并发处理能力和数据库性能是重点，特别是在游戏活动期间。安全性也非常重要，因为玩家数据和交易频繁，需要严格的数据加密和访问控制。

系统架构

着重于策略规划和资源管理，系统架构更注重数据的一致性和策略执行的准确性。可能采用更轻量级的客户端，服务器端则设计为支持大量并发的策略计算和资源调度，以及玩家之间的外交和战争系统。

技术架构

技术架构偏向于数据处理和分析，可能更依赖于大数据分析和AI算法来优化策略推荐和资源分配。服务器端可能更注重数据的一致性和高效查询。

部署架构

部署架构可能相对集中，因为SLG对延迟的要求不如MOBA严格，但同样需要考虑全球玩家的访问体验，可能会在关键地区部署节点。

1.3.3 MOBA

MOBA（多人在线战术竞技）游戏在服务器架构、网络延迟管理、数据同步与一致性、可扩展性与容错方面有其独特的要求和设计，这些方面往往区别于其他类型的游戏，如卡牌游戏或休闲游戏。以下是几个关键点的对比分析：

服务器架构

- 通常采用大服模式，即游戏服务按类型拆分，如大厅、战斗、地图等，采用集群部署，以支持高并发和可扩展性。
- 战斗服务器可能单独跨地域部署，以减少网络延迟，提升玩家体验。包含登录服务器、匹配服务器等关键组件，登录服务器通常利用弹性伸缩组避免单点故障，提高可用性。
- 游戏服务器（Game Server）负责处理游戏逻辑，可能采用高性能实例如c6.16x、c7.16x来保证游戏流畅度。游戏AI服务器可能需要更高配置如c7.32x、c8ae.16x以处理复杂的AI运算。

网络延迟管理

- 强调低延迟，因此在服务器位置选择上会更加考究，可能在全球多个地区部署服务器，确保玩家能够连接最近的服务器。使用全球加速服务，减少跨国数据传输的延迟。
- 同步方案倾向于帧同步，利用UDP协议为主，TCP作为备选，以优化网络通信效率。

数据同步与一致性

- 需要高度一致的数据同步，实时战斗数据处理，排行榜更新。尤其是在实时对战中，采用帧同步或状态同步等技术保证所有玩家看到的游戏状态一致。对于跨区域玩家互动，需要有高效的跨数据中心数据同步机制，以及处理脏数据的策略，确保游戏公平性。

可扩展性与容错

- 高度依赖于集群部署和弹性扩缩容能力，如使用ECS高IO服务器，确保在高峰期能够迅速增加资源，平滑应对流量波动。服务器无单点故障设计，利用ECS的自动宕机迁移能力，确保服务连续性。
- MOBA游戏在设计时更加强调低延迟、实时交互、数据一致性以及动态扩展能力，以适应其高强度的竞技性和大规模在线玩家的实时交互需求。

系统架构

以快速匹配、短时间高强度对抗为特点，系统架构强调低延迟和高度同步。通常有专门的匹配服务器处理快速匹配逻辑，游戏服务器则专注于实时战斗的处理，确保所有玩家间的动作同步无误。

技术架构

强调实时性和同步性，技术架构会采用帧同步或状态同步技术，确保游戏内的战斗体验流畅。UDP协议因其低延迟特性被广泛用于核心战斗逻辑的传输，同时辅以TCP来确保数据的可靠传输。

部署架构

部署架构强调低延迟，游戏服务器可能需要在主要玩家群体区域附近部署，以减少网络延迟。匹配服务器可能集中部署，而游戏服务器则分散部署，确保玩家能够快速匹配并进入游戏。

1.3.4 放置经营类

放置经营类游戏的系统架构、技术架构、部署架构的关键特征主要围绕其游戏特性展开，这类游戏通常对实时交互的要求较低，但强调长期的资源管理和策略规划。与动作类、射击类等对网络延迟敏感的游戏相比，放置经营类游戏的架构设计更注重数据的一致性、持久化存储、以及经济系统的平衡性。以下是几个关键特征及与其它游戏类型的区别：

系统架构

- 异步交互设计：放置经营类游戏往往允许玩家离线操作，游戏进程继续进行，玩家上线后获取离线收益。因此，系统设计需支持异步数据处理和同步机制，确保玩家操作与后台数据的一致性。
- 经济系统稳定性：这类游戏的经济平衡极为重要，系统架构需确保资源产出、消耗的算法稳定，防止经济膨胀或紧缩。
- 数据持久化与备份：由于游戏进度和资源积累对玩家来说非常重要，系统架构需重视数据的持久化存储和定期备份，确保玩家数据安全。

技术架构

- 重数据处理与分析：集成大数据处理和分析工具，用于玩家行为分析、游戏经济健康度监控，以及个性化推荐系统。
- 弱网络依赖：相较于实时竞技游戏，放置经营类游戏对网络实时性的要求较低，但仍然需要保证数据同步的准确性和效率，减少因网络问题导致的用户体验下降。

部署架构

- 多地域部署：虽然对延迟要求不高，但为了全球玩家的接入体验，可选择在主要玩家地区部署节点，减少网络延迟。
- 静态资源优化：大量使用图片和动画等静态资源，部署架构需优化CDN分发，确保资源加载快速，提升用户体验。
- 成本效益考量：由于游戏持续运行但不总是高并发，部署时需考虑成本效益，如使用按需付费的云服务，以及在低峰时段自动缩容以节省成本。

第二章 游戏网络：数字世界的神经与血脉

在数字娱乐时代，游戏网络如同虚拟世界的神经系统，承载着数亿玩家实时交互的每一次操作、每一帧画面和每一声语音。无论是MMORPG中千人同屏的史诗级团战，还是MOBA竞技游戏中毫秒级的技能判定，背后都依赖于复杂的网络架构实现全球玩家无缝连接。现代游戏网络需同时满足三大基础特性：

- **实时性：**MOBA游戏技能延迟超过100ms即影响胜负，VR游戏需保持20ms以内延迟防眩晕；
- **可靠性：**大型开放世界游戏每秒需处理200+实体状态同步，99.99%可用性是底线；
- **扩展性：**热门游戏上线首日可能面临用户量百倍暴涨，需分钟级扩容应对。

然而，全球分布式部署带来的跨国传输难题、玩家终端设备差异引发的网络抖动、黑客攻击导致的DDoS流量洪峰等挑战，让游戏网络成为技术攻坚的高地。因此，构建高性能游戏网络需聚焦三大核心诉求：

核心诉求一：合理架构——搭建全球化连接的骨骼

"节点部署如同围棋落子，既要抢占战略要地，又要形成全局联动"

- **覆盖广度：**头部游戏需在6大洲部署50+节点，确保全球玩家接入延迟<150ms
- **成本效率：**通过性能和架构优化降低30%成本，边缘计算减少40%回源流量
- **灾备设计：**多活架构实现故障30秒内自动切换，年故障时间<5分钟

核心诉求二：网络质量与加速——塑造流畅体验的肌肉

"玩家不会原谅卡顿，正如运动员无法容忍跑道塌陷"

- **传输优化：**QUIC协议降低弱网环境60%重连耗时，智能路由减少20%跨国绕行
- **网络加速：**跨地域/跨境网络加速，网络路径自动切换，实现0丢包，SLA保障99.99%可用性
- **安全防御：**T级DDoS清洗能力+AI对抗伪造攻击，保障游戏期间零中断

核心诉求三：智能运维——注入永续运行的基因

"运维不是在救火，而是在构建防火长城"

- **故障自运维：**自助诊断、发现问题，问题解决恢复时间提升50%
- **可观测性：**业务流量洞察，玩家状态可观测，单玩家路径追踪定位精度达99%
- **自动部署：**通过代码（IaC）工具实现网络自动部署，可靠运维和变更，加速部署流程30%。

2.1 合理架构——选点与网络规划

2.1.1 节点选型策略

全球分布式部署

根据用户分布选择不同Region覆盖用户，例如东南亚用户集中区选择新加坡节点，欧美用户使用法兰克福/弗吉尼亚节点，日韩用户使用日本节点。

国内选点建议

全局服，游戏平均延时要求100ms以内，可以选择上海或杭州地域覆盖全国 区域服，游戏平均延时要求50ms以内，可以选择北京覆盖华北、上海/杭州覆盖华东、广州覆盖华南、成都覆盖华中。

图1

海外选点建议

全局服，游戏平均延时要求200ms以内，可以选择新加坡/中国香港/美西覆盖全球 区域服，游戏平均延时要求100ms以内，可以选择日本覆盖东亚、新加坡/香港覆盖东南亚、法兰克福覆盖欧洲、弗吉尼亚覆盖美洲。

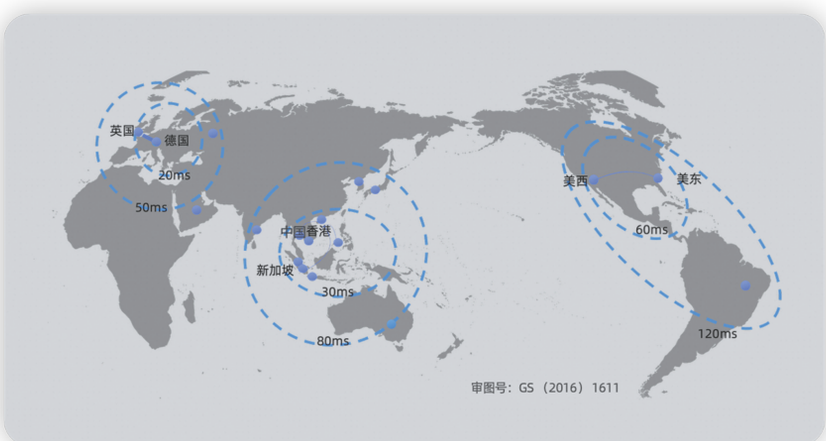

图2

2.1.2 网络拓扑设计

- **多活架构：**采用同城双活+异地灾备模式，例如在上海和深圳部署双数据中心，通过GTM(Global Traffic Manager)实现流量自动切换。
- **分层架构：**接入层（边缘节点）→ 逻辑层（游戏服务器集群）→ 数据层（数据库与缓存）分离，通过VPC实现安全隔离。
- **带宽预留与弹性伸缩：**基于历史峰值流量预留20%~30%带宽冗余，配合云服务的Auto Scaling实现动态扩容。

平台服网络最佳实践

平台服包含网关接入系统、逻辑匹配系统、数据存储系统、BI分析系统，服务体现为登陆服、排行服、社区服等；因为要承载所有玩家的访问，会存量极限场景下的大并发，一般都是集群部署，使用Load Balance作为集群入口，并按需配置公网入口和私网入口；

合理规划和选点

- 大陆和海外分别搭建平台服务，并结合玩家地域规划服务地域
- 基于业务时延需求选择集中或分区部署
- 使用阿里云性能观测服务辅助选点

设计网络架构

- 分层架构：业务分层分模块，使用ALB负载均衡构建集群，借助ALB丰富的转发规则能力实现业务搭建；
- 选择按量：通过共享带宽按量计费提供高性价比公网出入口；ALB/NAT高上限自动弹性灵活应对业务变化；
- 选择云原生：使用ALB实现SSL证书卸载，降低后端服务计算压力；使用ALBingress为容器集群提供高性能自动弹性7层转发能力；

设计加速

- 玩家网络：EIP多线BGP提供高质量公网；全球加速GA提升玩家体验；
- 运维网络：CEN实现全球高速稳定运维接入

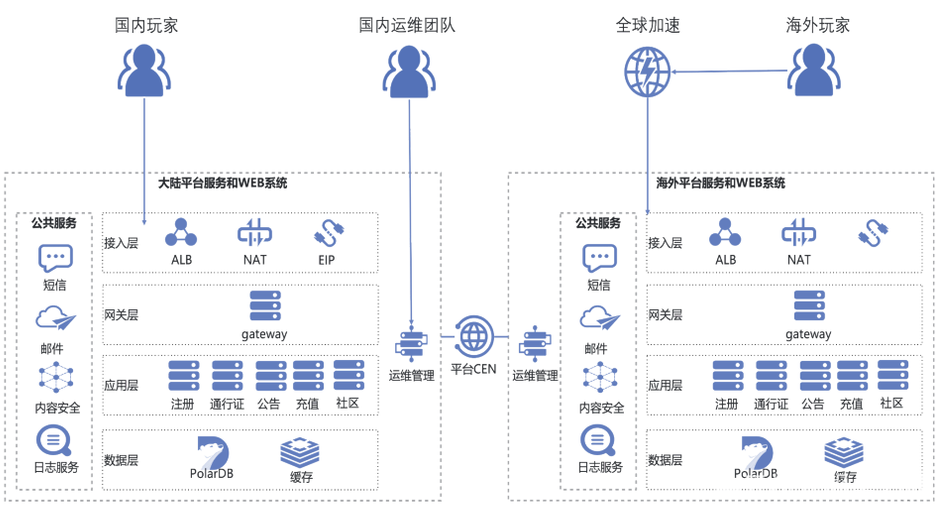

图3

游戏全球组网最佳实践

大服部署：一台高配服务器即一组Game server，可以承载很多玩家，甚至自带登陆、数据库缓存、排行等；

小服部署：低配服务器，数量较多，支撑单人或多人的战斗场景，玩家数据等会存放在统一的数据平台内；

集群部署：游戏服采用集群部署，多台ECS/容器pod挂载到一组Load Balance上，通过Port、URL等调度到不同的游戏房间内；

合理规划和选点

- 基于游戏时延基线合理规划游戏服地域
- 使用阿里云性能观测服务辅助选点

设计网络架构

- 选择按量：高防EIP保障游戏服平稳运行；通过共享带宽按量计费提供高性价比公网出入口；NLB弹性伸缩实现游戏服弹性扩缩架构
- 选择云原生：NLB可承诺99.995%服务可用性；NLB支持TCPSSL证书卸载；NLB CPS限速抑制连接风暴；NLB可以为容器提供高性能自动弹性4层转发能力

设计加速

- 玩家网络：多线BGP提供高质量公网；高防EIP提供抗D安全能力；全球加速GA提升玩家体验；云企业网CEN/全球加速GA保障跨区对战
- 运维网络：CEN帮助实现全球化高速稳定运维接入

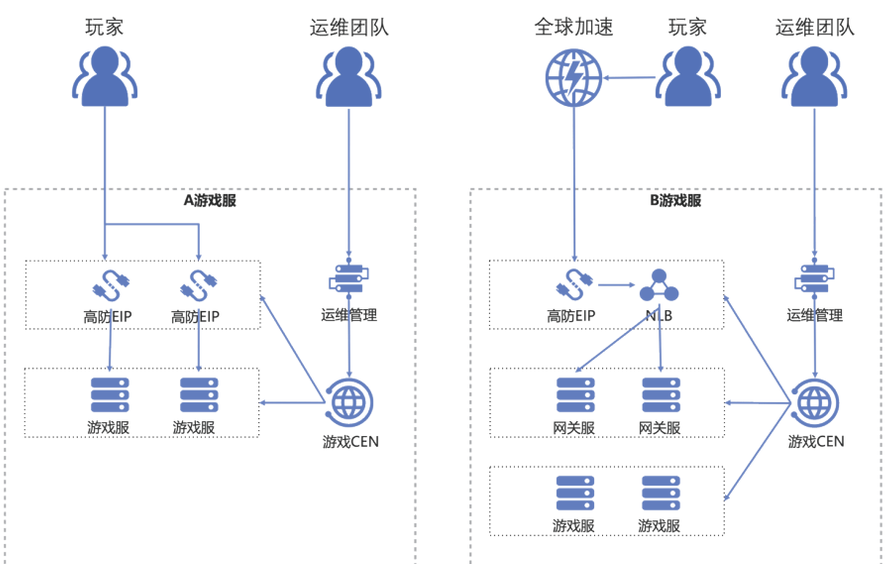

图4

游戏全球组网最佳实践

组网规划

- 全球一张网：通过CEN构建游戏全服务地域打通
- 网络隔离：基于业务流属性合理分割网络平面

- 访问控制：使用TR多路由表和前缀列表实现网络平面之间的精细化访问控制

服务之间网络优化

- 优化传输：服务之间互联互通的业务流量利用CEN线路资源实现全球高质量高稳定性互联互通
- 流量分级：平台服跨地域数据交互/游戏服跨地域运维/数据跨地区传输/平台服-游戏服业务互调/平台服-数据服数据传输等网络交互场景通过CEN多路由表/QoS能力实现平面内流量精细化管控

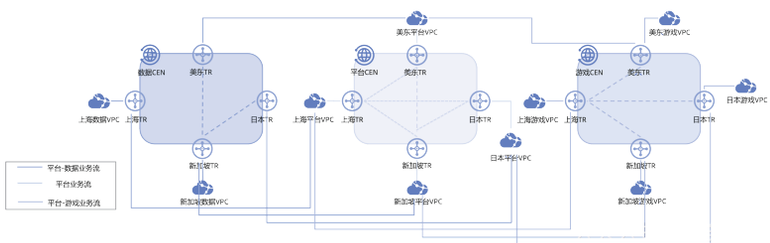

图5

2.1.3 网络高可用建设

公网/跨地域网络高可用

- 多线动态BGP可以实现AZ/Region级别容灾调度
- CEN主备环网，任意2地域之间不少于3条路径保护

网元高可用

- 全新NFV架构，双区双活，故障自动切换
- 控制面和转发面分离，均采用多节点部署

网络架构高可用

- CEN作为平台服与游戏服交互的主用链路，公网作为备份链路
- 主备链路通过域名实现切换
- 规划游戏内网域名和公网域名
- 规划平台内网域名和公网域名

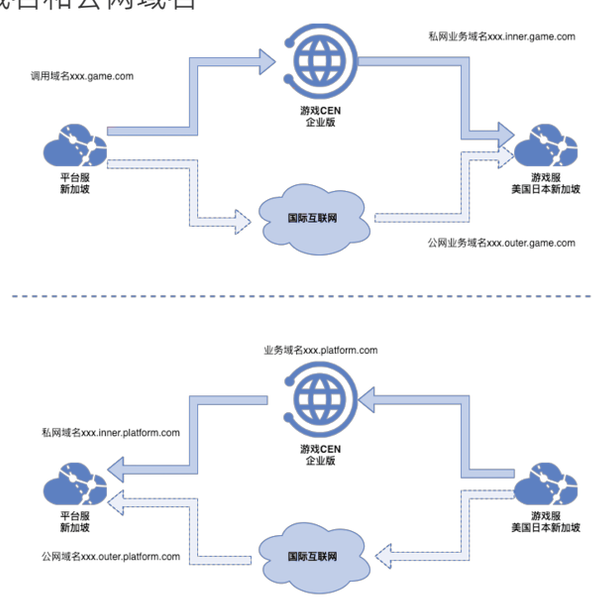

图6

2.2 网络质量与加速——玩家体验保障

2.2.1 游戏网络延迟需求

在游戏发展过程中，游戏厂商越来越注重游戏体验感。游戏体验感是指玩家在玩一款游戏时的综合感受，在具体细节上又包含多个方面，例如情感、社交、表达、交互、消费、宣泄、成长、挑战、探索、幻想、感官体验。其中最重要和最基本的就是游戏角色动作的流畅度，如果流畅度不够没有打击感，那么对于玩家而言，他们的游戏体验感就会很差。相反如果动作流畅、技能华丽、连招可以无缝连接、操作手感和打击感爆棚，那么玩家的游戏体验感就会很好。影响游戏流畅度除了客户端和服务端性能外，最重要的就是网络质量，如果网络存在频繁抖动、丢包、拥塞，必然会影响游戏流畅度。所以很多游戏厂商在游戏架构设计之初就为了提高网络稳定性和可靠性做了很多努力来保证游戏用户的流畅度体验。

游戏网络加速大体分为如下几个场景：

- 游戏国内发行强烈依赖广电总局所发放的游戏版号，如无版号，只能在海外发版，并通过加速器产品覆盖国内玩家
- 游戏出海后，东南亚基础设施落后，部分游戏无本地覆盖条件，只能中心化部署，存在加速诉求
- 游戏客户对时延敏感，特别是球类、FPS即时对战类游戏
- 客户业务分布在多AZ机房，不同AZ距离公网出口距离不同，如上海地域嘉定等地的机房离物理公网出口较远
- 游戏服部署在边缘区域，平台服部署在中心区域，游戏服与平台服之间传输追求极致最短时延

2.2.2 游戏加速器（跨地域隧道）

场景介绍

早期网游研发和发行商大多是国外公司，国内玩家玩这些游戏的时候经常会出现卡顿、延迟情况，这时候游戏加速器应运而生。游戏加速器本质上并不是让游戏的运行速度加快了。游戏的运行速度，除开硬件配置、带宽的因素外，出现卡顿、延时的原因还有不同运营商的网络壁垒、不同地点不同国家服务器的设立等待、运营商跨境链路拥塞和抖动，而游戏加速器就是把这些原因避开，就像堵车的时候单独开出一条专线道路，同时还用导航告诉你最近的路径，这样数据传输的路径变得畅通，距离缩短，从而从感官上体验到速度加快了。

目前市场上常见的游戏加速器有迅游，奇游，海豚，UU等等，主流的加速技术实现原理无外乎2种：LSP和VPN。前者也可以称为socket5代理，后者可以细分为PPTP,L2TP,OPENVPN等VPN技术。

部署架构

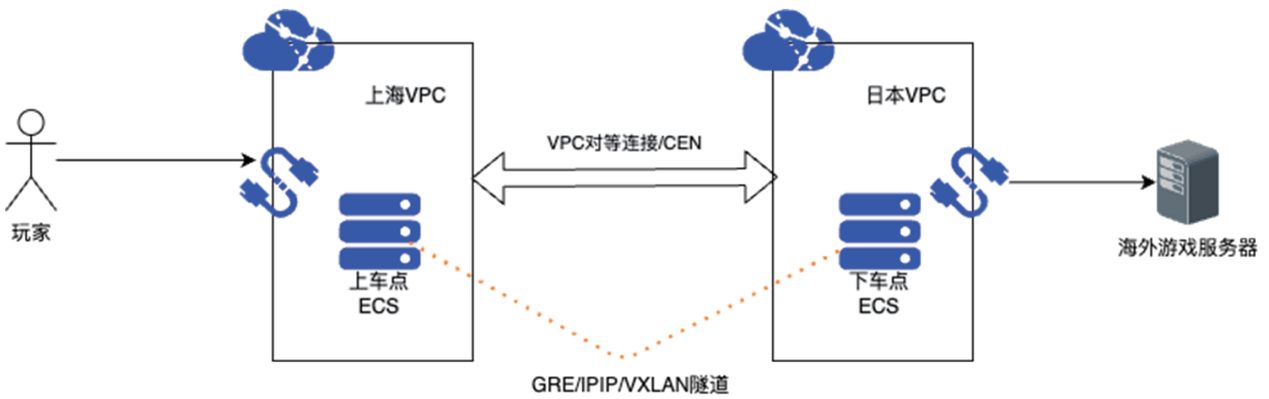

图7

加速原理

这里以VPN模式为例，介绍加速器在云平台的路由转发实现方式：

- 玩家通过公网VPN连接到上海加速器节点ECS（业内称上车点），连接VPN后生成虚拟网卡，分配VPN规划的IP网段，同时在玩家客户端植入到游戏服的IP地址段路由从VPN虚拟网卡出。
- 在上海ECS上面加游戏服IP地址路由指向IPIP、GRE、VXLAN隧道接口发给日本ECS（业内称下车点）
- 日本ECS接收到去往游戏服的数据包从日本本地EIP出去前往日本游戏服务器
- 日本ECS完成VPN客户端虚拟IP地址的SNAT功能，同时回指客户端IP路由到隧道接口发给上海ECS，再由上海ECS给客户端回包。

此方案通过阿里云内部专线提供跨地域/跨境互通，可以有效避免运营商骨干网/跨境网络出现抖动、拥塞的情况，提升游戏网络稳定性。

2.2.3 东南亚回国加速（精品EIP）

场景介绍

东南亚华人文化圈分布广泛，人口已达2400万人，而且东南亚华人对访问中国互联网游戏，视频/短视频平台有较强诉求。国内游戏企业也纷纷出海期望能在东南亚游戏市场占领一定份额。在加上近年来，国家管理机构为了规范游戏的健康发展，游戏版号发放收紧，很多游戏厂商会选择在东南亚发布游戏服务，覆盖东南亚玩家。如果一款游戏在东南亚能够火爆，也会吸引很多国内玩家，但是国内玩家访问东南亚游戏服务器、视频平台质量不佳，网络不稳定，在不增加太多成本的情况下需要一个方案来优化国内玩家访问东南亚游戏服务器。

部署架构

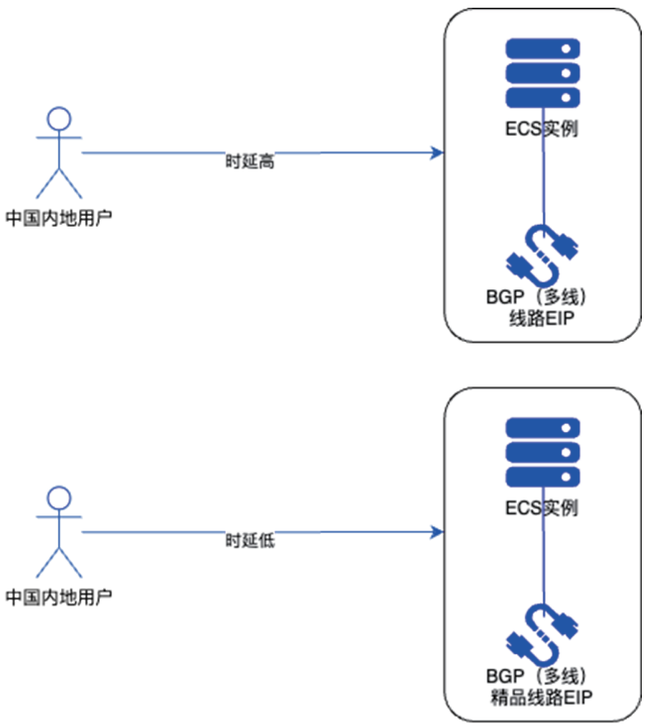

图8

加速原理

国家针对中国香港和中国内地范围内的电信运营商牌照资质要求不同，导致中国香港和中国内地之间的公网互访资源有限，我们通过划分不同EIP产品的优先级，来满足不同用户的需求；普通EIP不能确保用户数据包到中国内地的路由走向的，完全依赖运营商互联网路由协议的实际情况，所以普通EIP真实可能会出现绕行日本、美国在回国的情况。针对需要低延时公网返回中国内地的用户，可购买精品回国EIP，保证用户使用合法的最优公网路径。目前支持精品EIP的地域有中国香港、日本（东京）、新加坡、马来西亚（吉隆坡）、菲律宾（马尼拉）、印度尼西亚（雅加达）和泰国（曼谷）。

精品EIP的底层原理其实比较简单，精品EIP底层同时具备3大T（电信(CN2)、联通精品、移动）的优质回国线路能力，会根据运营商实际水位情况有一定的调度，默认情况下：

- 中国内地 -> 中国香港： 流量无需跨运营商网络，直接通过终端对应3大T的网络访问中国香港服务器；
- 中国香港->中国内地： 由于联通和移动容易拥塞，默认流量先走电信和联通精品网络再到对应运营商网络；
- 当某一运营商出现拥塞/故障的情况下，会主动调度，将流量分摊到其他空闲运营商网络，最大程度保障线路质量；

2.2.4 游戏跨境/跨域加速（全球加速GA）

场景介绍

游戏服务器部署在海外阿里云站点或者其他云站点，需要加速国内的用户和海外的用户访问海外游戏服务器，游戏同服或者区域同服的场景。由于游戏部署地域限制，游戏服务器

无法在所有用户地域或者国家部署，或多或少会存在跨地域或者跨国访问的情况。特别是东南亚等地区运营商的基础网络建设较差，如果依赖公网跨境访问游戏服务器就无法避免的会出现比较差的网络质量。

部署架构

图9

加速原理

全球加速不仅可以加速阿里云的服务，只要是源站有公网访问能力的都可以通过全球加速实现网络加速。全球加速网络覆盖全球范围，并且和云安全联通，加速的同时降低游戏服务被攻击风险。不管游戏服务器部署在阿里云站点还是其他站点，都可以使用网络加速服务。而且通过全球加速能够获取用户客户端的真实源IP地址，可以让游戏厂商分析真实玩家信息。

游戏厂商将业务域名解析到全球加速的CNAME中，然后在全球加速上配置各个地域的上车点，当玩家访问游戏服务器的时候，不同地域的玩家会自动解析到就近上车点完成访问接入。然后再上车点和下车点之间的网络是阿里云高质量的全球互通网络，可以很好的保证传输质量，避免网络抖动和丢包。用户添加对应的下车点后，全球加速会在对应的地域部署下车点资源，使请求就近下车访问游戏服。

下面是某客户从国内访问澳洲服务的加速效果对比，通过全球加速，客户的网络质量和用户体验都有了显著的提升。

直接访问源站，没有加速的效果

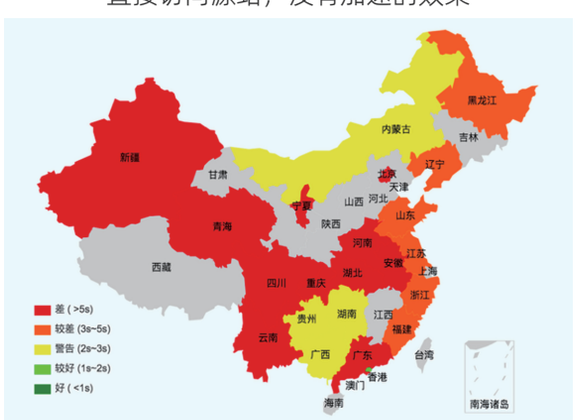

图10

GA加速后的效果

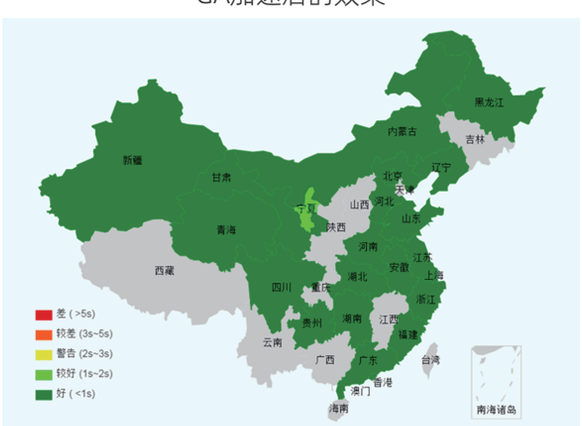

图11

2.2.5 游戏网络延迟需求

场景介绍

游戏国内发行强烈依赖广电总局所发放的游戏版号，如无版号，只能在海外发版，并通过加速器产品覆盖国内玩家。游戏出海后，东南亚基础设施落后，部分游戏无本地覆盖条件，只能中心化部署，存在加速诉求。同时部分场景全球加速GA无法满足游戏厂商需求，因为GA是类似代理转发的模式，无法在加速服务器上做二次开发，例如认证等行为，客户需要在跨地域/跨境加速的同时在加速服务器上做一些二次开发的内容。此时EIP IP target就可以比较好的满足此类客户的需求。

部署架构

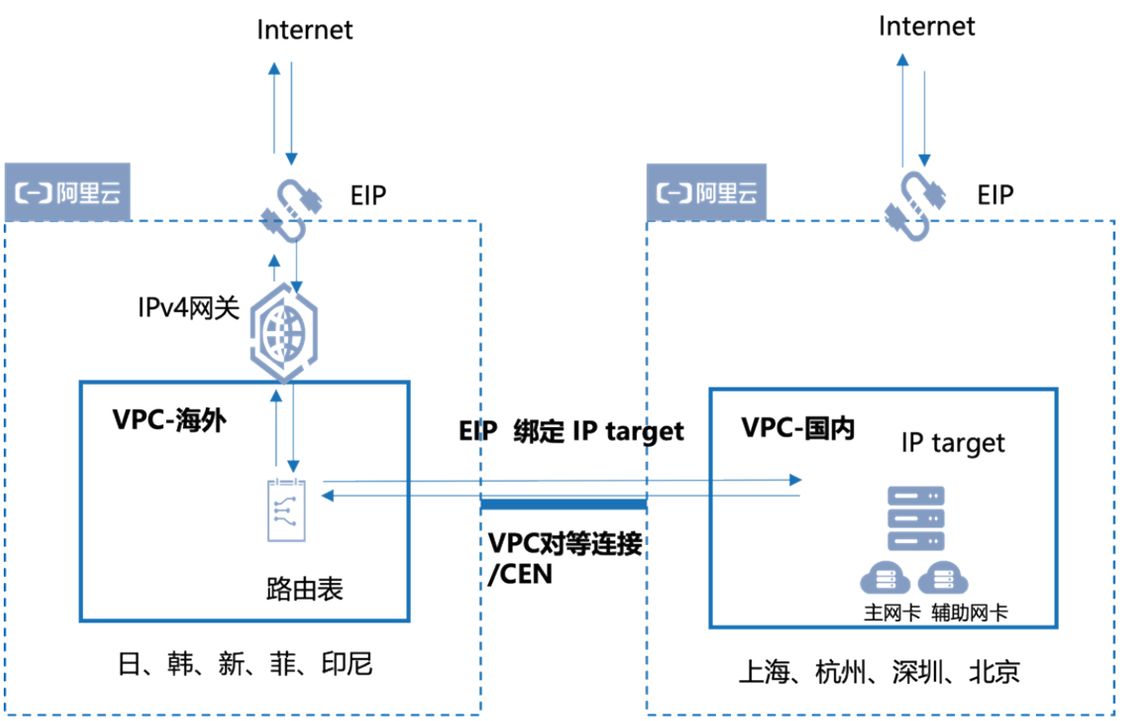

图12

加速原理

EIP ip target加速原理，就是将EIP绑定到跨地域的ECS服务器上，访问EIP即相当于访问跨地域的云服务器，不需要经过跨境、跨地域运营商骨干网传输。具体实现方式如下：

- 购买东南亚Region的EIP，例如购买新加坡EIP。
- 配置EIP后端服务器的内网IP，如配置为上海服务器的内网IP。
- 在新加坡地域VPC路由表中配置与上海VPC互通的路由，两地域互通可以通过VPC对等连接，也可以通过云企业网CEN。
- 在上海服务器内配置多网卡，辅助网卡绑定新加坡EIP，主网卡绑定上海本地域EIP。
- 在上海VPC配置默认路由，当上海地域ECS响应数据给新加坡的时候，也是经过对等连接或者CEN
- 响应报文经过新加坡VPC后转发给IPV4网关转发会给新加坡当地游戏客户端。

解决方案优势

- 通过EIP IP Target功能远端挂载ECS，帮助客户降本
- 通过VPC Peer CDT按量计费，用多少付多少
- 支持按量付费的同时，支持按CEN按带宽付费
- VPCPeer/CEN，提供优质线路匹配低时延诉求

2.2.6 海外游戏服加速（Anycast EIP）

场景介绍

中小游戏厂商在游戏发布前期基于成本考虑只会把游戏服务器部署在少量地域，但是游戏玩家分布在不同地域或者国家。如果玩家访问游戏服务器需要跨境或者跨运营商会产生比较大的网络波动、延迟，影响游戏的流畅度。游戏厂商期望寻找一个不增加太多成本和技术改造的前提下优化玩家网络。

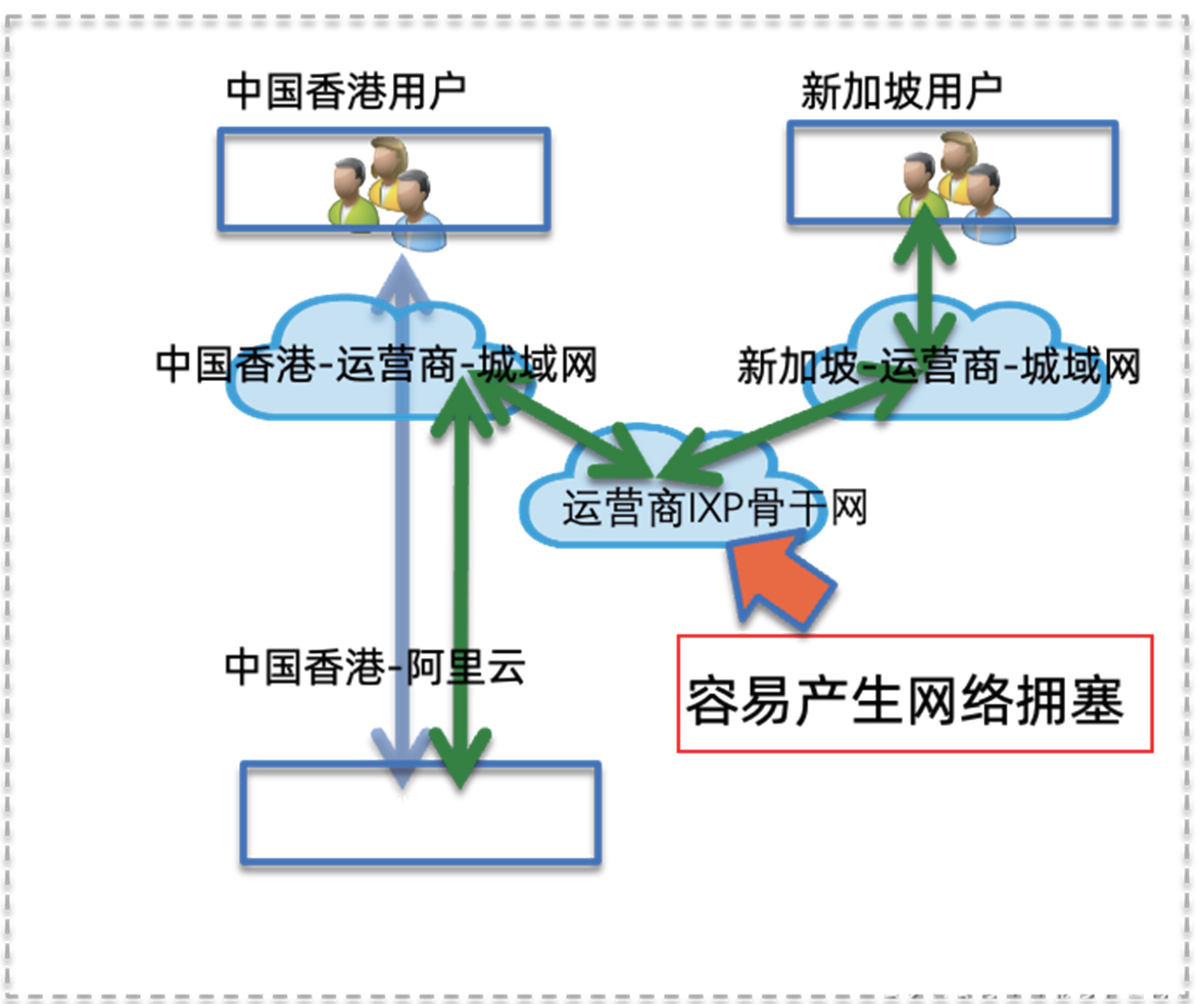

图13

部署架构

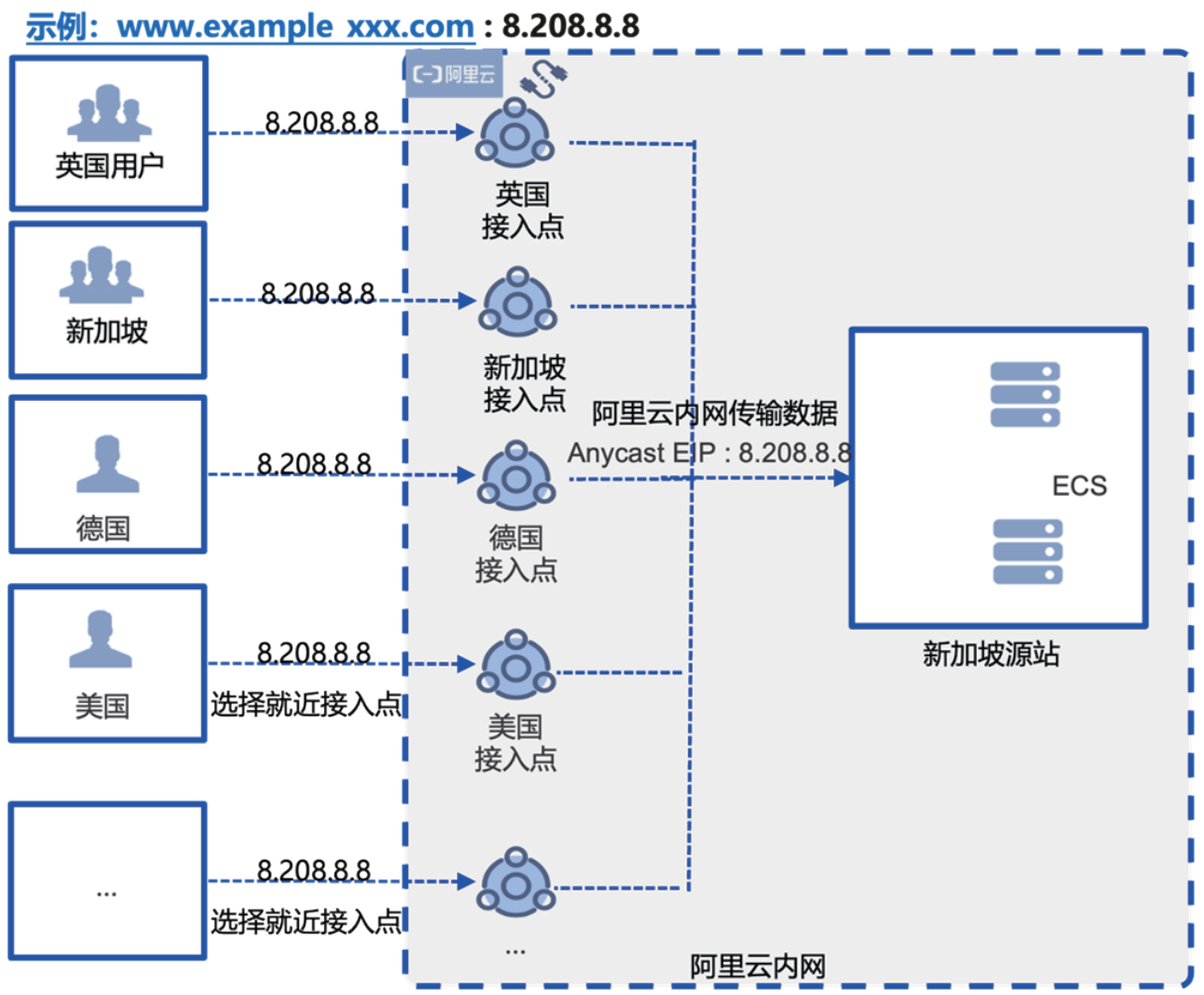

图14

加速原理

Anycast 是一种网络寻址和路由技术，它允许多个设备共享同一个 IP 地址。使用 Anycast，一个 IP 地址可以被分配给多个设备，但在任意时刻只有一个设备会被路由到并处理来自客户端的请求。当一个客户端向共享 IP 地址发送数据包时，这个数据包会被发送到最近的设备，通常是最短网络距离的设备，从而实现了负载均衡和容错。Anycast EIP利用阿里云高可靠低延迟全球传输网络，玩家就近接入阿里云网络，将互联网用户的流量传输到您部署服务的地域。

使用Anycast EIP时：

- **入站流量：**优化入口位置，来自互联网的入站流量将从距离来源最近的接入点进入到阿里云的全球传输网络。
- **出站流量：**出站流量经由阿里云全球传输网络从距离目的用户最近的接入点传出。这种路由方式尽可能使用阿里云高可靠低延迟的全球传输网络，从而最大限度的降低网络拥塞，提高网络性能。

2.2.7 公网可用区加速（指定AZ申请EIP）

场景介绍

游戏客户大多对对时延敏感，部分游戏类型例如球类、FPS、MOBA等即时对战类游戏更是对每一毫秒延迟都关注。游戏客户端访路径总体如下，客户端经过家庭宽带或者移动网络接入运营商，运营商通过骨干网络请求转发到阿里云公网接入点，公网接入点在讲请求转发到对应服务器。如下以手游为例，网络延迟会在如下几个地方产生

① 玩家手机终端到手机基站的延迟
② 手机基站到运营商骨干网的延迟
③ 运营商骨干网到阿里云公网接入点的延迟
④ 阿里云公网接入点到游戏服务器的延迟

①②③这三部分的延迟是想对固定的，④的延迟取决于阿里云公网接入点与阿里云服务器之间的距离。在海外地域中，每个地域的可用区当前最多3个可用区，各可用区到公网出口Pop点机房的时延差异不大。但是在国内地域中，通常两个可用区作为物理公网出口，各个可用区距离公网出口可用区延迟不同，离的越近延迟越低。又由于目前EIP申请的时候是随机分配可用区的，而且用户层面无法感知EIP是在哪个可用区，所以EIP会与服务器不在同可用区导致④这一部分的延迟不是最优的，可能会比同可用区场景下高几毫秒。

部署架构

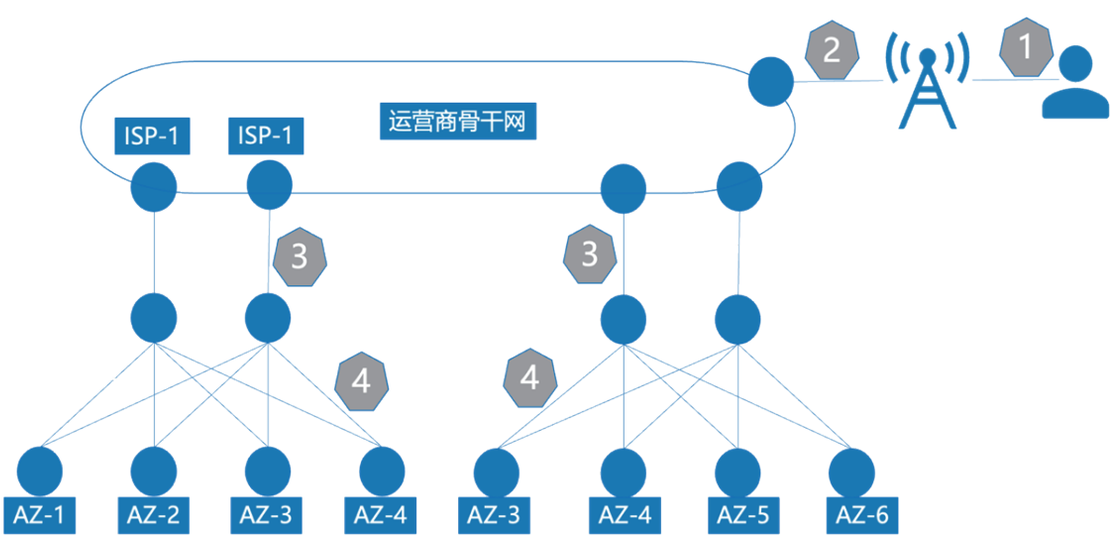

图15

加速原理

结合ECS资源、不同AZ与物理公网出口的距离，指定可用区申请EIP网段，使EIP与服务器在相同可用区，最大化减小阿里云公网接入点到服务器之间的延迟。如果涉及共享带宽，也需要将共享带宽迁移到离公网最近的可用区。通过这种方案优化后可以加速几毫秒的公网访问延迟。

2.2.8 跨地域内网加速（CEN铂金）

场景介绍

部分游戏会有全球同服的部署架构，游戏服和中心服会通过云企业网打通，跨地域互通对时延敏感。游戏加速器行业提供的加速服务，也是端到端的网络延迟有极致需求，需要提供强有力的网络能力才能在行业中占有一席之地。所以在游戏场景中往往存在需要优化跨地域延迟的情况。阿里云云企业网虽然可以保证跨地域互通的稳定，避免丢包和抖动。但是跨地域互通是存在冗余负载链路的，也就是说两个地域互通可以通过多个不同物理路径到达，当一条路径失效后自动切换到另外一条路径。此方案虽然提升了可用性，但是会存在部分流量经过低延迟路径，部分流量经过高延迟路径。如果想要优化这个延迟就需要让流量都先优选低延迟路径进行转发以保证跨域的最短路径低时延。

部署架构

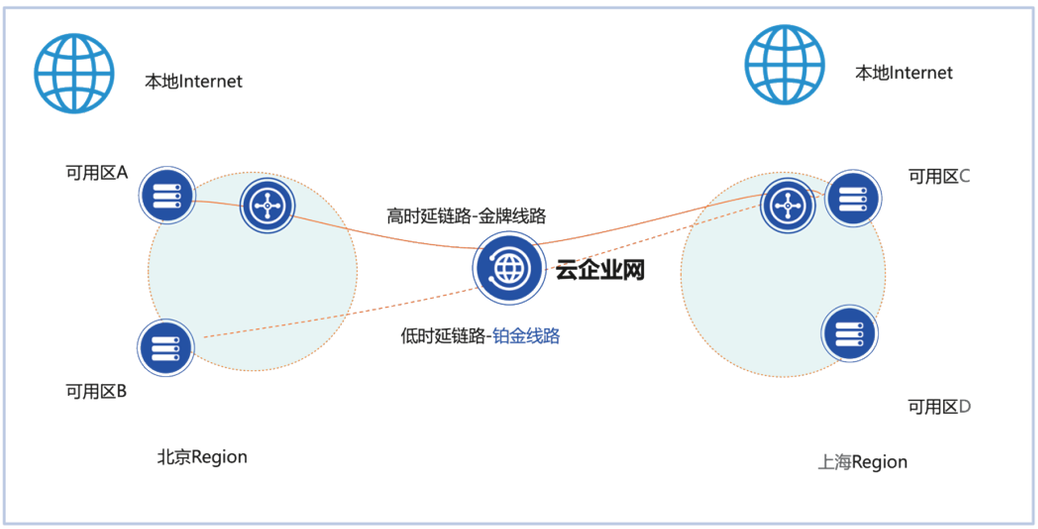

图16

加速原理

阿里云CEN/VPC对等连接的跨地域互通使用的是阿里云的骨干网（也叫ABTN骨干网），阿里骨干网可以根据IP报文头的DSCP打标，选择某些特定链路或者优先级队列，根据DSCP提供不同等级的服务质量。

在跨域访问场景下，将跨域报文打上特定的DSCP标，根据用户实例和地域（cen+src-region+dst-region）绑定dscp，让ABTN骨干网提供相应等级的服务质量。在CEN场景中，如果需要极致优化延迟，就可以将对应客户的CEN流量打上铂金的标签，使其跨域互通的流量能够优先最低延迟路径。由于最段路径的带宽是有限的，带宽资源也是稀缺的，只能优先保证确实对延迟极其敏感的用户使用。

常用的质量等级有：

- 金牌：默认，默认服务
- 铂金牌：高等级，提供更短的RTT，即短延迟和低抖动，成本高
- 铜牌：低等级，当网络设备发送拥塞时，先丢铜牌的报文

2.2.9 方案对比

表2

	跨地域隧道	精品EIP	全球加速GA	EIP IP Target	Anycast IP	指定AZ申请EIP	CEN金牌
适用场景	跨地域/跨境加速	东南亚跨境加速	跨地域/跨境加速	跨地域/跨境加速	海外跨境加速	云内可用区之间加速	云内跨地域加速
费用	高	中	高	低	中	低	中
配置复杂度	高	低	高	高	中	低	低
业务改造	高	低	高	中	低	低	低
优点	自建灵活，根据业务时间不同需求	配置简单	适配UDP/TCP/HTTP等业务，支持IDC/任意公网IP源站	可以根据业务调整进行二次开发	配置简单	配置简单	配置简单
缺点	配合也业务改造均比较复杂	只支持亚太地域	费用想对较高，需要对业务改造	配置想对较复杂	只支持境外不支持国内	无	无

2.2.10 客户案例

2.2.10.1 全球同服- 某客户全球同服案例

客户需求及背景

- XXX老牌游戏厂商，SLG类的小游戏为主，业务集中部署在阿里云XX region，对全球玩家提供服务；
- 需要保证业务部署的可靠性；
- 需要保证全球玩家访问的时延可控；

方案设计

- 稳定性考虑：源站采用集群部署模型，SLB作为集群入口，后端real-server分散到不同的可用区，提升跨可用区的容灾能力；
- 弹性能力：根据玩家的数量动态扩容SLB后面的ECS数量，并通过prot调度；通过SLB数量增加，横向扩容游戏源站的规模；
- 玩家加速：前面部署全球加速GA，采用智能路由方式，将玩家访问流量吸流到阿里云内网，走加速路径，匹配不同的域名送到不同的源站SLB。

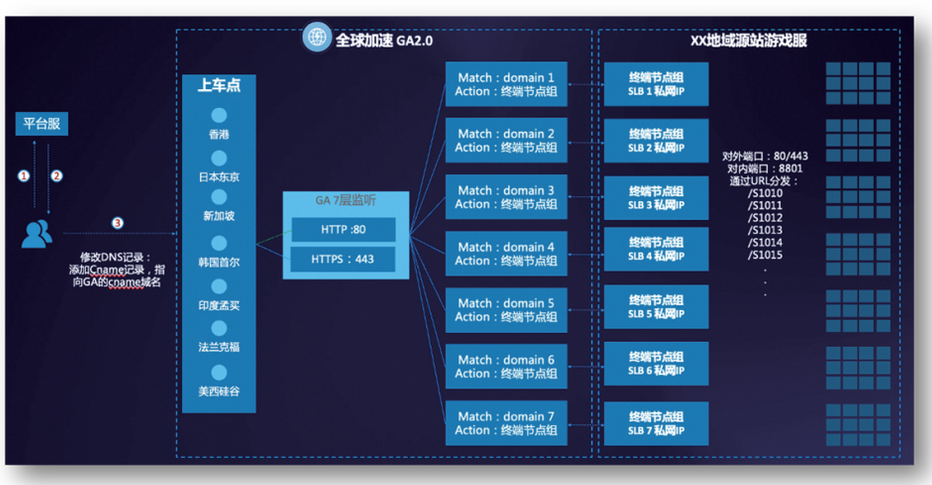

图17

业务价值

游戏源站高可靠性：通过SLB主备集群部署，以及游戏服多可用区部署，把容灾能力做到最大。

玩家游戏极致体验：根据玩家分布按需创建上车点，canme智能解析，让玩家走最快最稳定的线路回源。

2.2.10.2 某客户 Anycast EIP+CEN标杆案例

客户背景

XX网络是国内老牌游戏厂商，一直坚持精品战略，自主研发了《征途》系列、《球球大作战》等深受玩家喜爱的精品游戏。《球球大作战》更是开创休闲竞技全新游戏品类，累计用户量6亿。以往在国内自发游戏只选择在IDC，海外只发行过少量轻量级游戏，一直排斥和各大云厂商在公共云方面的大规模合作。

业务需求

- 游戏出海 - 此次新游《我们的派对》是首个在海外发行的重度游戏，希望选取的云厂商在游戏领域具备深厚沉淀
- 玩家体验 - 作为海外新游戏诚意之作，确保海外玩家网络高质量接入成为重中之重

网络方案

- 游戏服 - 新加坡Anycast EIP源站覆盖海外各国的游戏玩家就近接入，构建区域同服
- 平台服 - 通过云企业网CEN互联国内和海外的平台服业务，高质量的跨境云专线快速构建全球一张网

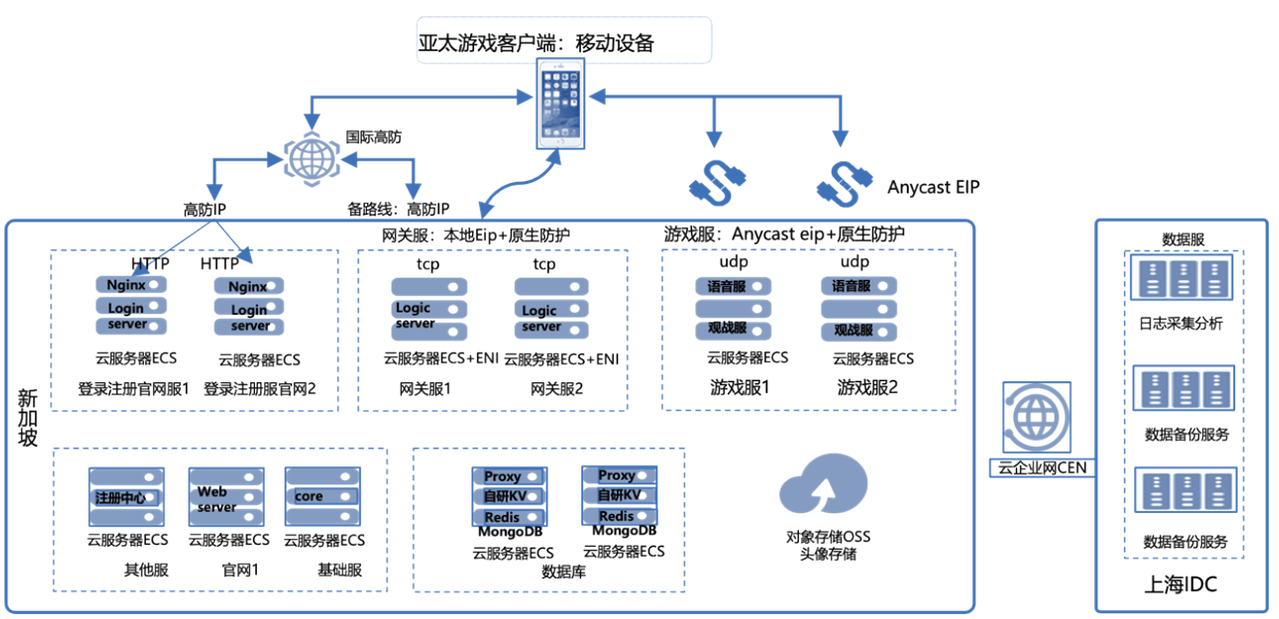

图18

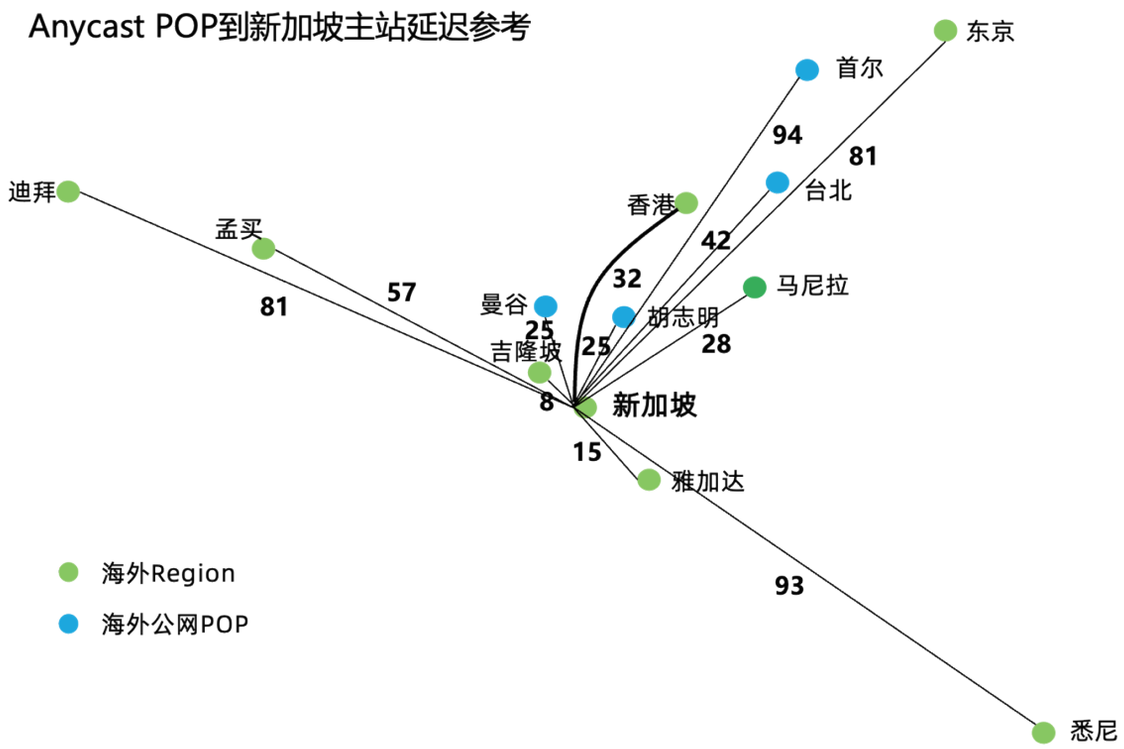

图19

业务价值

- 加速布局海外 - 亚太领先的公网质量，跨境合规的CEN，帮助客户快速评估并部署目标节点
- 提升游戏体验 – Anycast EIP助力业务就近接入，解决亚太各国跨境公网线路拥塞问题，大幅降低了卡顿率和重连率，提升玩家体验

2.2.10.3 某客户-GA自定义帮助游戏房间访问加速

客户背景

XX客户MOBA类游戏，玩家匹配成功进入新房间，完成一局战斗。

游戏房间动态创建，每台server上会承载多个房间，使用不同的进程，对外服务的port不同，但是不同server上的port可能会重复，玩家会通过唯一的IP+port进入到匹配好的房间内。

业务需求

- 玩家可能来自不同的城市，在游戏过程中访问游戏服的距离不同，网络时延和稳定性也存在差异，公网的不确定性会影响玩家的游戏体验，希望能有网络加速的接入，改善玩家的体验；
- 会有一些玩家出国旅游，反馈访问效果差，如果能够安全合规完成加速；

方案设计

- 使用全球加速GA的自定义路由模式进行加速调度：
- 对后（game servers）：使用加速port，屏蔽不同game server上相同port的影响；
- 对前（玩家）：返回给玩家端的仍然是一组IP+port，不改变玩家的正常访问逻辑；

- 加速调度：在房间匹配的逻辑服上做加速开关，随时可以进行切换

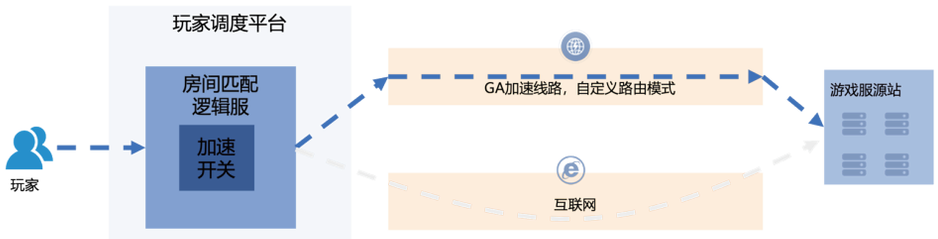

图20

客户价值

- 操作简单：对现有业务架构的改动较小，只需在原有的调度逻辑服上维护一下加速映射表；
- 效果明显：玩家就近接入阿里云全球加速POP点，全程走阿里云内网回到源站，稳定、可靠、时延低；
- 安全合规：大部分链路在阿里云内网，数据更加安全；
- 全球互通，跨境使用联通合规的跨境专线，无合规风险；

2.3 智能运维--高效网络管理

游戏网络运维在游戏运营过程中有很多痛点，如何在复杂的网络环境中更好的运维网络是个很重要的命题。

1.复杂网络架构的管理难度高

游戏网络通常采用全球化分布式架构，涉及多地域节点、混合云部署及边缘计算，运维人员需同时管理VPC、负载均衡、跨地域专线等复杂组件，手动配置易出错，且难以实时监控全局状态。例如，跨国玩家接入时可能因路由策略不当导致延迟激增，影响体验。

2.网络质量波动影响玩家体验

实时对战类游戏对延迟敏感（要求<50ms），但跨国传输易受网络抖动、丢包影响。传统运维工具难以快速定位链路问题，例如某条国际线路拥塞导致玩家卡顿，需人工逐层排查，耗时长达数小时。

3.故障定位与恢复效率低

突发故障（如DDoS攻击、服务器宕机）可能导致大规模玩家掉线。传统日志分析依赖人工经验，无法快速关联告警与根因，例如某次服务中断可能由安全组误配置、数据库过载或CDN节点异常等多因素叠加引发。

4.混合云与多云协同挑战

游戏企业常采用混合云架构（如核心数据在私有云、计算资源在公有云或者使用多个云厂商提供游戏服务），但跨云网络配置复杂，流量调度策略难以统一，且运维工具缺乏跨平台整合能力，导致资源利用率低下。

5.安全防护与合规压力

游戏行业是DDoS攻击的重灾区，传统防火墙难以应对T级流量洪峰；同时，数据跨境传输需符合多地法规（如GDPR），网络策略需动态调整以满足合规要求。

为应对游戏网路运维中的痛点，我们可以使用阿里云产品来提升游戏网络运维能力。

2.3.1 自动化运维体系

NIS（Network Intelligence Service）巡检功能通过提供云网络架构的可观测服务，帮助客户实现云网络运维的自服务和智能化，主要体现在以下几个方面：

2.3.1.1 问题可视化

NIS巡检功能能够精准发现云网络中的异常，如BGP连接异常、EIP未绑定实例、TR冗余链路未配置等问题，并通过直观的五维图表展示风险项。这使得客户能够一目了然地了解网络架构中存在的问题，从而快速采取行动。

2.3.1.2 故障自运维

NIS巡检不仅指出问题，还提供了详细的优化建议。例如，当检测到BGP连接异常时，NIS会提供具体的资源ID和地域信息，帮助客户快速定位问题。客户可以根据NIS的建议，自行处理问题，无需等待外部支持，大大提高了运维效率。

2.3.1.3 流量分析和管理

NIS可以进行流量的分析与洞察，可以查看公网流量分析统计和内网流量分析统计，统计维度可以分为实例级别（一元组），IP级别（二元组），端口应用级别（五元组），访问源（国家/城市/ISP）。通过流量分析可以感知到游戏玩家流量分布，以及快速定位异常的玩家情况。比如通过公网流量分析可以知道公网是否有被攻击，是否有某个地域/省份有网络波动和丢包等异常。通过内网流量分析可以知道应用之间的交互情况，当出现异常的时候可以排查各服务器或者组件之间交互的网络流量情况，来分析和定位问题。

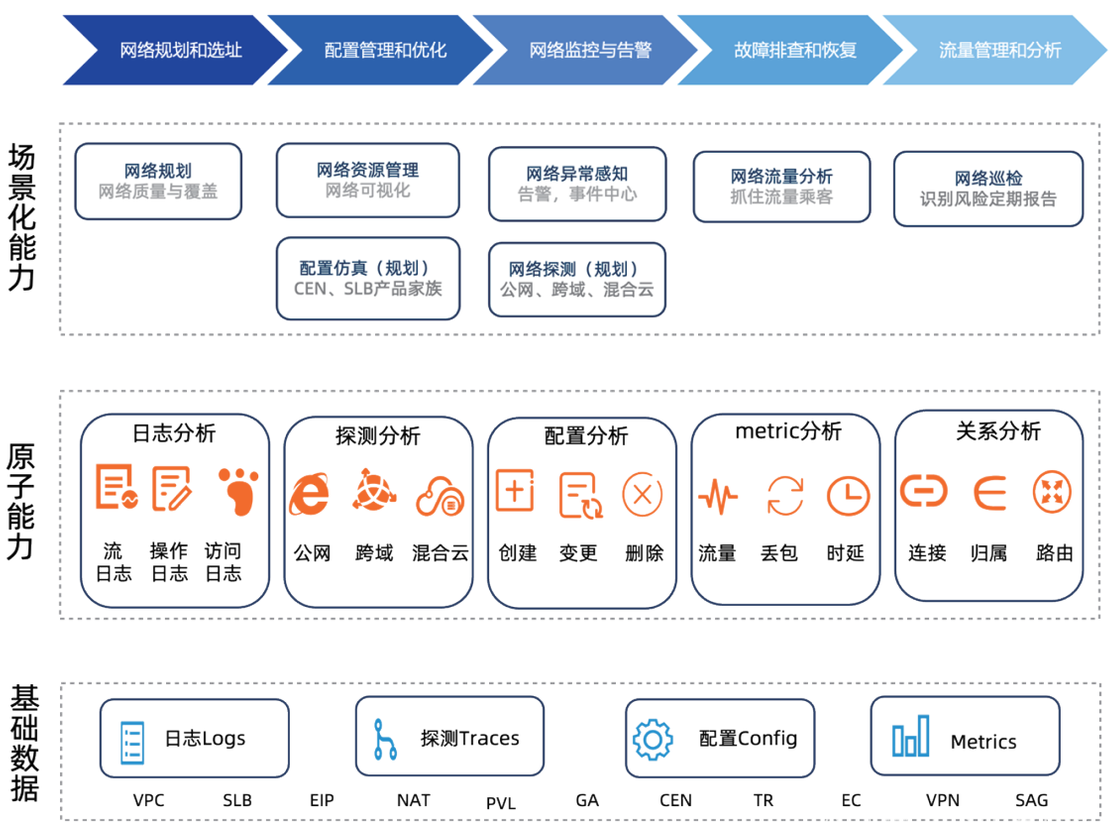

图21

2.3.2 网络质量监控

2.3.2.1 游戏网络监控难点

游戏场景中，通常对网络质量要求较高，网络抖动和延迟都会影响玩家体验，甚至是区域性网络异常短时间内无法快速恢复会严重影响游戏体验，严重的情况下还会影响客户游戏登录、支付、战斗等场景。玩家对游戏卡顿往往容忍度极低，如果是少量玩家反馈网路异常也不确定是否是玩家本地网络问题,例如家庭宽带导致游戏体验。

在网络问题排查过程中，一般需要收集ping,telnet，双向mtr等信息来判断物理链路层面是否正常，以及通过这些信息判断异常的中间节点。但是目前绝大多数游戏都是手机端游戏，手机端不像电脑端可以便捷的执行ping/telnet/mtr等命令，必须安装第三方软件进行信息收集，而且还需要特定玩家在异常场景下收集信息。这种情况下一是比较难找到可以配合收集信息的玩家，二是采集的信息可能滞后，问题可能异常恢复。

即使有时通过第三方拨测如听云、博睿等，由于拨测节点有限，无法覆盖真实玩家所有地域和运营商，通过拨测可能无法发现问题，也无法判断问题点。

2.3.2.2 基于网络质量分析器监控

基于游戏网络中的这些痛点，可以使用网络质量分析器来进行问题分析和发现。

网络质量分析器是一款针对真实终端用户网络质量性能进行分析的SaaS服务。可以将网络质量分析器的SDK插件集成到App中，实时感知所有App真实用户在线情况、访问互联网的网络质量情况。网络质量分析器提供从在线终端设备发起到目标服务器的网络质量探测分析，帮助终端用户进行网络问题排查诊断。

同时网络质量分析器提供Android、iOS等版本SDK插件，支持各种操作系统版本，适配兼容性强；从真实终端用户采集的网络质量数据，客观还原边缘终端用户网络状况；支持常见的HTTP、MTR、TCPPING、PING等网络探测协议，满足日志采集需求。

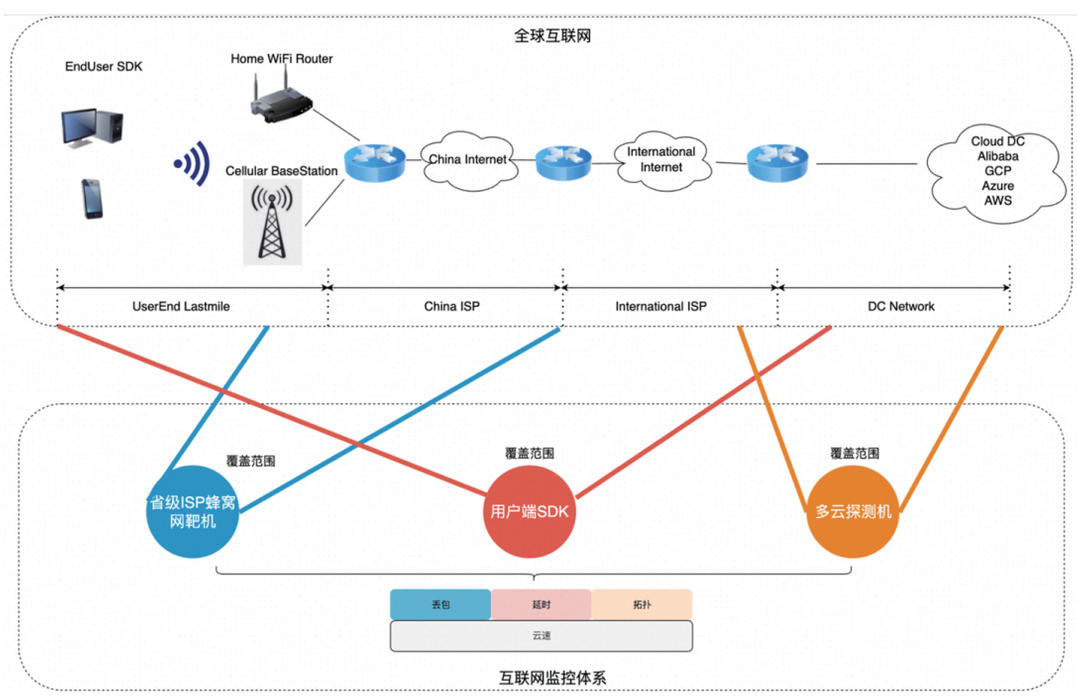

图22

网络质量分析器通过SDK形式集成在游戏终端上，当异常发生的时候可以通过日志查看对应全链路信息，即使是单用户问题也可以快速判断问题发生原因以及问题发生于哪个节点，然后可以进行对应恢复方案和处置。

图23

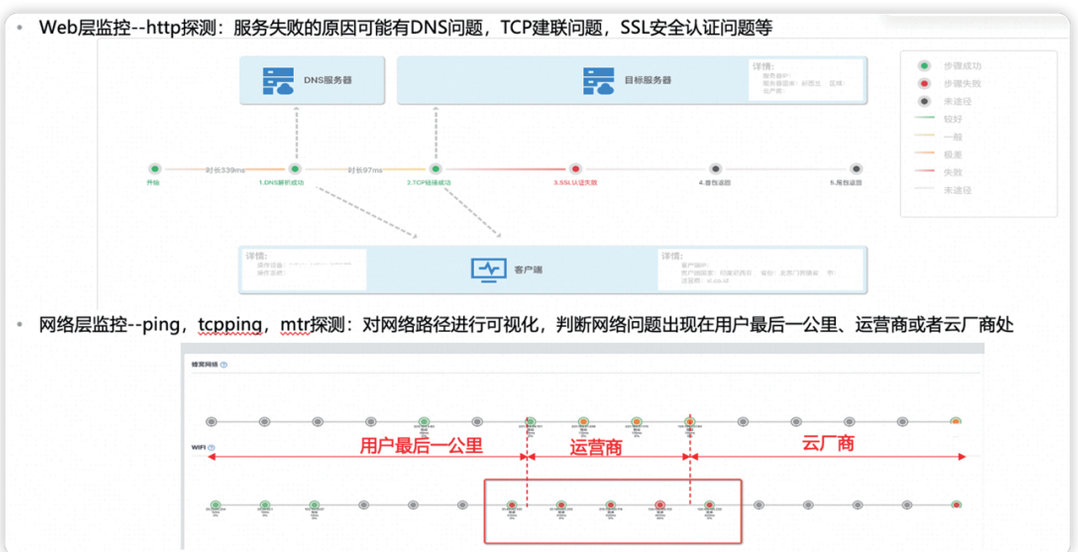

图24

2.3.3 Terraform构建自动化网络

在游戏网络场景中，使用Terraform进行自动化网络部署和运维能够显著提升效率、减少人为错误，并确保环境的一致性。Terraform是一个开源的基础设施即代码（IaC）工具，它允许用户通过编写配置文件来定义和提供数据中心的基础设施。

2.3.3.1 Terraform在网络运维中的优势

1. 基础设施即代码（IaC）

- 版本控制：所有的基础设施都可以像应用程序代码一样被纳入版本控制系统中。
- 可重复性：相同的配置可以在不同的环境中多次应用，确保一致性。

2. 提高生产力

- 快速部署：只需几分钟即可完成复杂网络结构的搭建，大大缩短了开发周期。
- 易于维护：修改基础设施变得简单快捷，无需人工干预或复杂的命令行操作。

3. 增强的安全性和合规性

- 审计跟踪：所有对基础设施的更改都被记录下来，便于审查和追踪。
- 权限管理：可以精确控制谁有权访问和修改特定的资源。

4. 跨平台支持

- Terraform支持多种云服务商（AWS, Azure, Google Cloud等），甚至包括本地数据中心。这使得混合云或多云策略成为可能，增加了灵活性。

5. 社区支持和扩展性

- Terraform拥有庞大的社区支持，提供了丰富的插件和模块库，帮助简化常见任务的实现。

总之，在游戏网络场景下使用Terraform不仅可以加速网络部署流程，还能有效降低运维成本，同时提高了系统的可靠性和安全性。通过这种方式，团队可以更专注于核心业务逻辑的开发，而不是基础设施的管理。

2.4 游戏网络技术展望

2.4.1 Serverless架构：无服务器化降低运维复杂度

Serverless架构通过全托管服务和事件驱动模型，将游戏开发者从底层基础设施的运维中解放，实现零服务器管理、按需付费和毫秒级弹性伸缩。这一模式尤其适用于实时对战、全球同服等场景，能够显著降低运维复杂度并提升资源利用率。以AWS GameLift为代表的托管服务，已成为游戏网络架构演进的重要技术路径。

目前，阿里云未推出与AWS GameLift完全对标的专为游戏设计的全托管Serverless服务。AWS GameLift的核心功能（如全球分布式服务器部署、智能匹配引擎、自动弹性扩缩容）在阿里云生态中需通过多产品组合实现。阿里云Serverless产品家族（如函数计算FC、Serverless应用引擎SAE、弹性容器实例ECI）已具备支持游戏网络Serverless化的能力，尤其在弹性计算、事件驱动、无服务器化运维等场景中表现突出。

表3 关键能力对标AWS GameLift

AWS GameLift功能	阿里云替代方案
全球服务器部署	全球加速GA + SAE多地域集群部署
FlexMatch智能匹配	函数计算FC + 表格存储OTS自定义规则引擎
混合云支持（GameLift Anywhere）	SAE + 云企业网CEN + 本地IDC容器化部署
全托管运维	SAE自动化运维 + 日志服务SLS + 应用实时监控ARMS
按需计费	SAE按vCPU/内存使用量计费 + 函数计算FC按调用次数计费

2.4.2 边缘云游戏：结合ENS边缘计算实现云游戏10ms级延迟

随着游戏行业的快速发展和技术的进步，边缘云游戏架构正逐渐成为游戏网络演进的一个重要趋势。这种架构不仅能够解决传统游戏网络中的诸多痛点，还具备一系列独特的优势，预示着其在未来有着广阔的发展前景。

边缘云游戏解决的痛点

- 降低延迟：对于实时互动要求极高的游戏来说，低延迟是至关重要的。通过将计算资源部署在靠近用户的边缘节点上，可以显著减少数据传输的距离和时间，从而有效降低网络延迟，提升用户体验。
- 提高稳定性：传统集中式云计算可能会因为单点故障或网络拥堵而导致服务中断。而边缘云游戏则可以通过分布式的边缘节点来分散负载，即使某个节点出现问题，其他节点也能继续提供服务，增强了系统的稳定性和可靠性。
- 优化带宽使用：边缘计算允许更多的处理工作在本地完成，减少了需要传输到中心服务器的数据量，这不仅降低了对骨干网带宽的需求，也减轻了核心网络的压力。
- 增强沉浸感：借助边缘云的强大计算能力和低延迟特性，开发者可以创造出更加复杂、逼真的游戏环境和体验，如虚拟现实(VR)和增强现实(AR)游戏，为玩家带来前所未有的沉浸感。

边缘云游戏的优点

- 灵活性与可扩展性：边缘云游戏架构支持快速弹性扩展，可以根据实际需求动态调整资源分配，确保无论用户数量如何变化都能保持良好的服务质量。
- 成本效益：相比传统的数据中心解决方案，边缘云减少了长途数据传输的成本，并且由于资源利用效率更高，整体运营成本得以降低。
- 个性化体验：基于地理位置的服务使得根据不同地区的用户偏好定制化内容和服务变得更加容易实现，进一步提升了用户体验。

发展趋势与未来方向

展望未来，边缘云游戏预计将沿着以下几个方向发展：

- 技术创新：随着5G技术的普及以及硬件性能的持续提升，边缘云游戏将能够支持更高分辨率、更流畅的游戏体验。同时，AI和机器学习算法的应用也将使游戏内容更加智能和个性化。
- 生态建设：为了促进边缘云游戏的发展，构建一个开放、协作的生态系统至关重要。包括游戏开发商、运营商、硬件制造商在内的各方需共同努力，推动标准制定和技术共享。
- 跨平台整合：未来的边缘云游戏不仅限于单一设备或平台，而是要实现PC、主机、移动设备之间的无缝切换，让玩家随时随地享受高质量的游戏体验。

边缘云游戏凭借其在降低延迟、提高稳定性等方面的显著优势，正在重塑游戏网络的未来格局。随着相关技术的不断进步和生态系统的日益完善，我们有理由相信，边缘云游戏将会开启一个全新的娱乐时代。

第三章 游戏安全：守护虚拟王国的坚固壁垒

3.1 游戏业务中不同网络安全架构及其对应挑战

在游戏业务中，网络安全设计是保障用户体验和业务稳定运行的重要环节，围绕业务侧的安全需求展开，同时兼顾网络性能的优化表现。然而，由于不同类型的游戏项目（如：回合制、开放世界、多人在线竞技等）以及不同的业务场景（如登录验证、实时对战、社交互动等）对网络的需求各不相同，因此在实际设计中会衍生出多种技术特点鲜明的网络安全架构方案。这些架构不仅要应对常见的安全威胁（如DDoS攻击、数据安全、游戏作弊），还需确保低延迟、高并发等要求以满足业务性能要求要求。这种多样化的业务需求也带来了复杂的安全挑战，例如：如何在安全与性能之前做出平衡取舍、如何快速响应新型攻击手段等。

以下简要介绍游戏行业中常见的3种网络安全架构、对应的安全挑战及常用解法：

3.1.1 架构1：业务主机直接向公网暴露端口

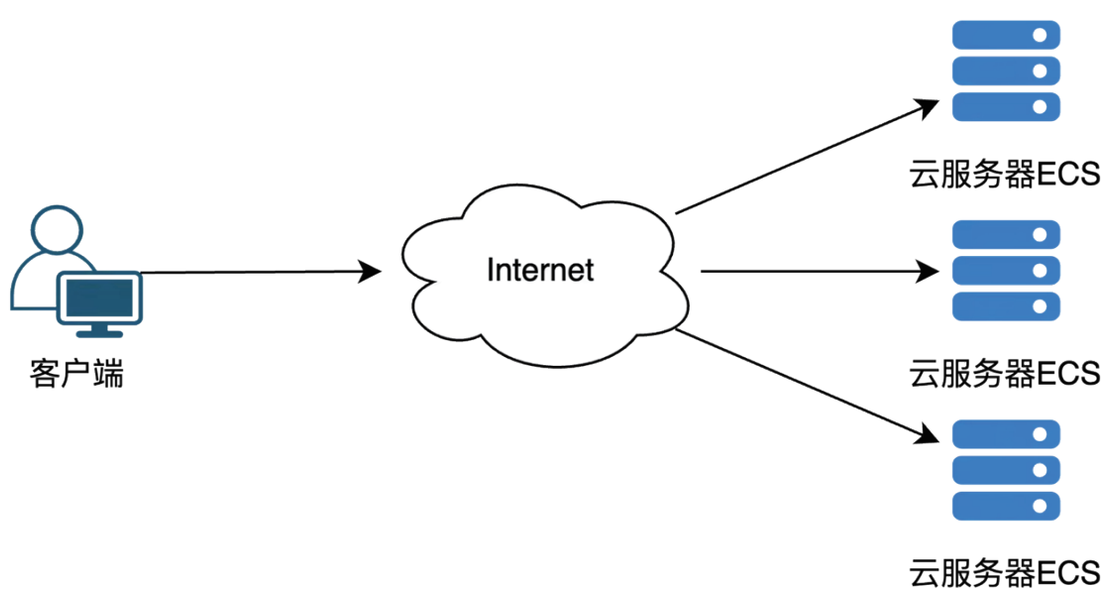

图25

常见业务场景

- 游戏网关类业务，如：网关服务、大厅服务
- 游戏战斗业务，如：大世界探索、副本联机

架构特点

- 业务端口主要以四层业务为主；且游戏行业中UDP端口业务占比相比其他行业偏高
- 业务主机规模较多，单游戏区服主机数目两位数以上，大型游戏项目则会直接暴露三位数以上的端口

- 业务端口为自定义端口居多，不使用常见端口（如80/8080/443/8443）对外提供服务
- 业务协议类型以自定义协议居多，不使用常见协议（HTTP/HTTPS）对外提供服务
- 客户端在前置业务流程中获取到准确IP后，直接通过公网IP进行访问，不经过域名解析
- 由于客户端与服务端之间不经过额外转发节点/转发节点极少，整体业务耗时偏低，适合部分时延敏感业务场景（如MOBA、FPS）

安全挑战

- 需要使用有限的安全资源同时防御大量主机业务端口
- 业务流量特征较为多变，难以使用标准规则防御针对此类业务的CC攻击，主要以防御流量型DDOS攻击为主
- 针对此类业务的小带宽高PPS攻击较为隐蔽，需借助时序告警及事件告警进行识别

常用解法

- 使用云原生防护企业版（国内地域）或者高防EIP（国内、海外地域）对主机公网业务进行防御

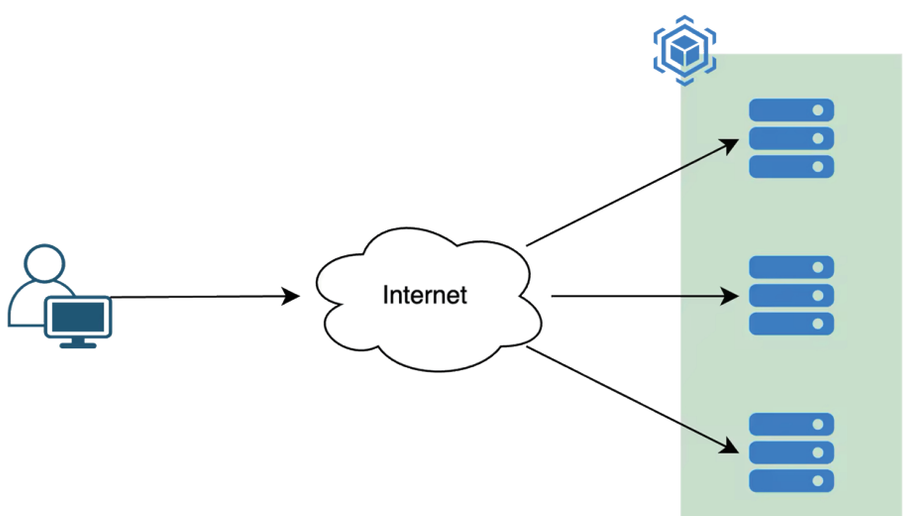

图26

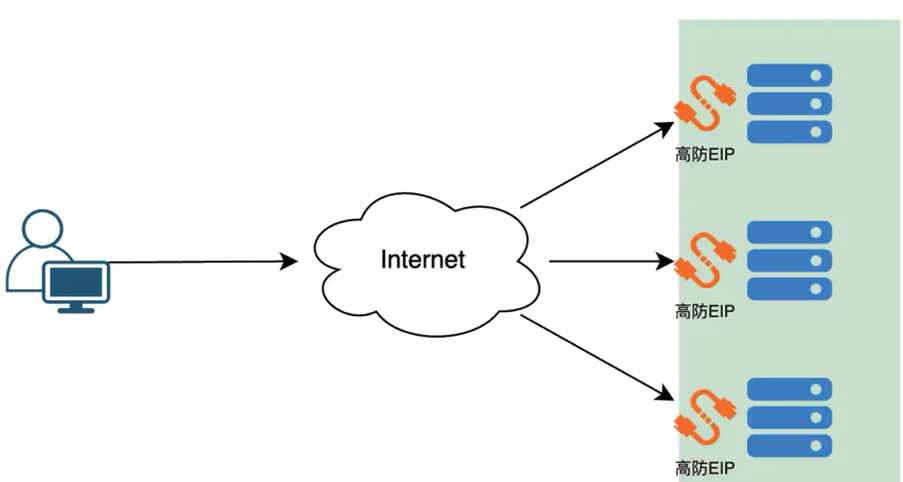

图27

3.1.2 架构2：业务集群向公网暴露四层业务接口

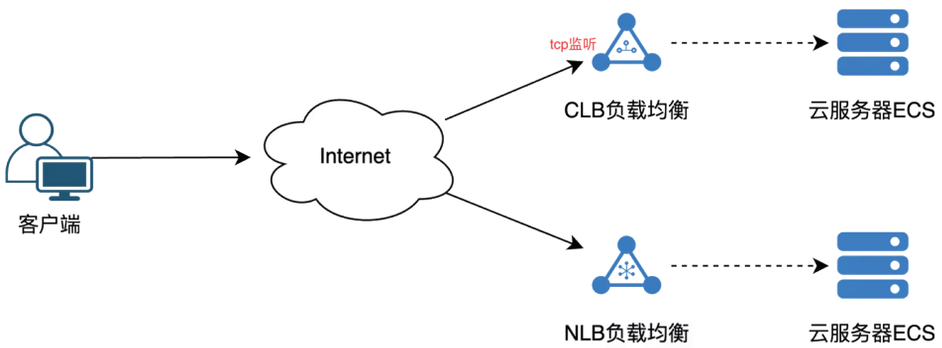

图28

常见业务场景

- 游戏网关类业务，如：网关服务、大厅服务
- 游戏鉴权业务，如：登录鉴权

架构特点

- 业务端口协议主要以TCP为主
- 源站常见为四层负载均衡（CLB四层监听/NLB），少量业务因特殊技术需要源站为主机
- 业务源站数目较少，一般来说两位数源站可基本满足单一游戏区服需求
- 业务端口多变，常见端口（如80/8080/443/8443）和自定义端口均有应用
- 不同的业务端口可能复用相同源站资源
- 客户端通过业务域名进行访问

安全挑战

- 业务流量特征较为多变，难以使用标准规则防御针对此类业务的CC攻击，主要以防御流量型DDOS攻击为主
- 发生CC攻击时（如恶意占用服务端连接数），需要同时借助安全服务和源站的能力予以应对

常用解法

- 使用新BGP DDOS高防（国内地域）或者国际DDOS高防（海外地域）对四层公网业务进行前端防护

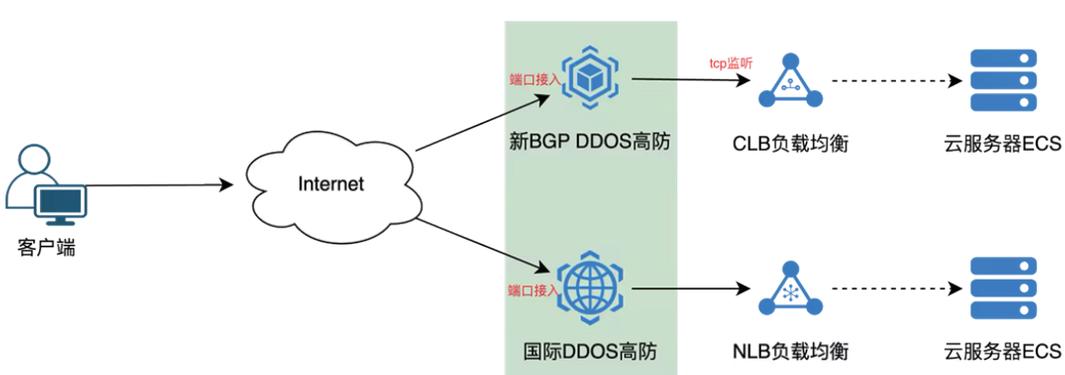

图29

3.1.3 架构3：业务集群向公网暴露七层业务接口

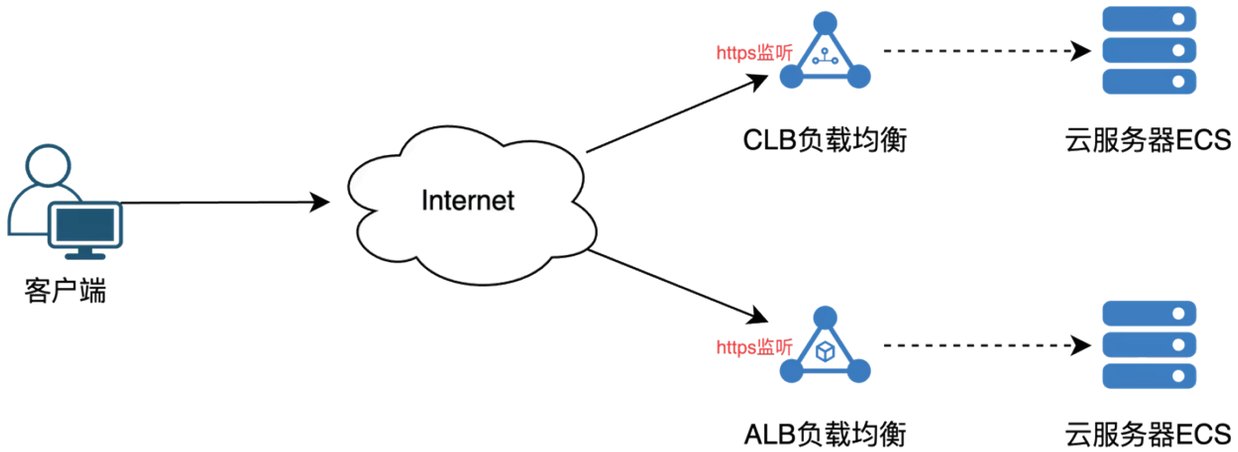

图30

常见业务场景

- 游戏2C的API接口，如：账号注册、登录鉴权、区服调度、游戏充值
- 游戏核心流量入口，如：塔防类游戏、棋牌类游戏、或者其他对网络延迟有一定容忍度的游戏类型
- 用户增长相关业务，如：游戏官网、游戏论坛、游戏官方商城

架构特点

- 业务端口协议主要以HTTPS为主
- 源站常见为七层负载均衡（CLB七层监听/ALB），对整体业务时延有一定容忍度
- 业务源站数目较少，一般来说两位数源站可基本满足单一游戏区服需求
- 业务端口较为集中，主要为443/8443等HTTPS端口
- 在承载API类业务时，业务并发连接数与请求数比例在1:5左右（即单个TCP连接处理5次HTTP请求）；在承载交互频次较多的游戏互通流量时，业务并发连接数与请求数比例可达到1:10甚至更高
- 客户端通过业务域名进行访问

安全挑战

- 业务流量规模与游戏业务安排正相关，在游戏活动、新游开服、商城促销等场景中会有明显的集中性访问，对全链路可靠性有较高要求
- Web服务信息在公网中曝光量巨大，攻击者可轻松获取到目标信息进而发起各类攻击，受攻击概率更高
- 同理，攻击者针对Web服务的攻击类型也相对较多且规模较大，DDOS流量带宽最高可达到Tbps级别

常用解法

- 使用多种代理安全服务（新BGP DDOS高防、国际DDOS高防、Web应用防火墙WAF、边缘安全加速ESA）等对七层公网业务进行综合防护

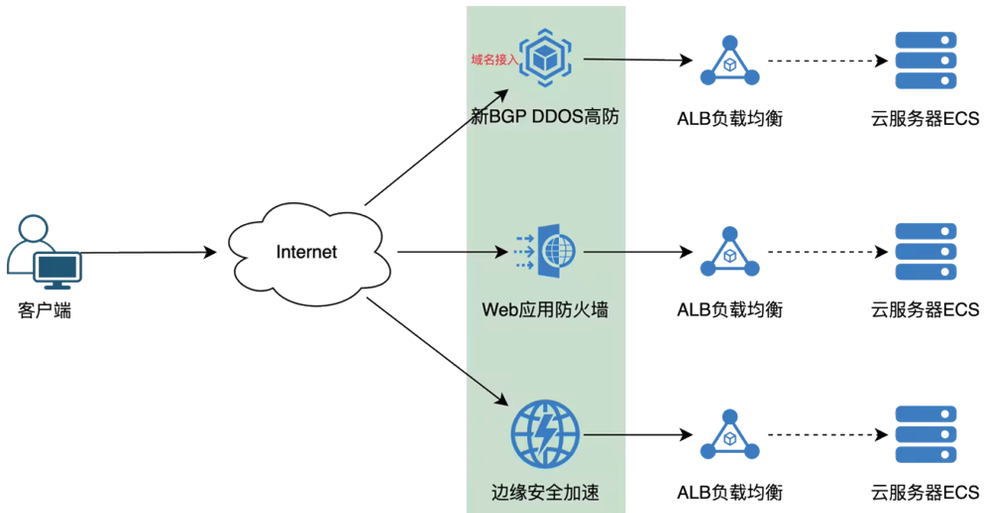

图31

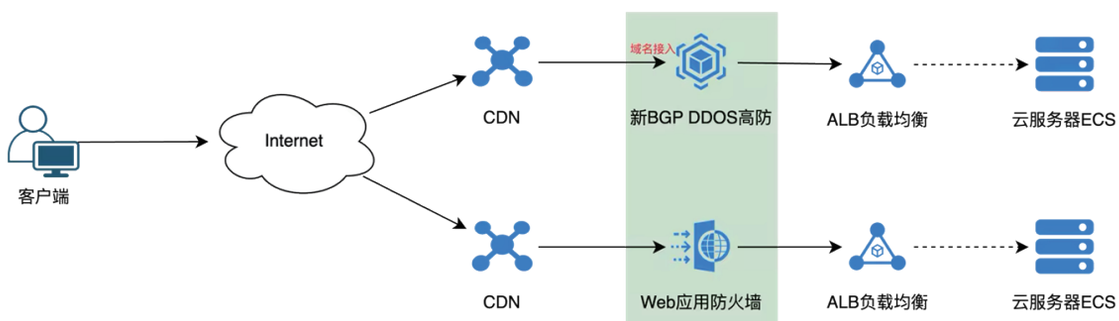

图32

3.2 如何应对游戏业务中网络安全挑战

3.2.1 秘籍1：高冗余水位防护

网络安全威胁具备隐蔽性高、来源分散、量级不可控的特点，为降低异常流量过高导致业务可用性受损的风险，需要在游戏业务链路中进行高冗余水位防护，用相对充足的防护资源换来更可靠的网络安全防护表现。具体可从下述三个方面进行设计：

1.云安全服务高冗余设计

- 适当预留安全服务实例资源，为业务运行过程中常规应急预留资源调度空间，如：防护资产数、业务带宽、防护带宽、业务QPS
- 优先考虑启用云安全服务弹性扩容能力，应对业务突发请求和异常流量突增场景，提高安全服务容错能力
  - DDoS原生防护：支持弹性业务带宽
  - DDoS高防：支持弹性业务带宽、弹性业务QPS、弹性防护带宽
  - Web应用防火墙：支持弹性QPS

2.源站资源高冗余设计

- 源站为负载均衡类（CLB、NLB、ALB）时，使用2个或以上负载均衡实例承载相同业务流量，使得网络入口具备高冗余能力
- 源站为IP类（EIP、ECS PIP、公网NAT）时，使用共享带宽或者按量付费模式提升单个IP的公网吞吐上限，使得网络入口带宽有一定资源冗余

3.业务逻辑与性能高冗余设计

- 为单个业务节点的性能资源设计冗余，如：性能测试中测得单台网关服务器可容纳玩家数上限为2000，在实际业务上线后设置其告警上线为1500人，预留一定资源冗余
- 为业务集群的性能资源设计冗余，如：根据运营数据设计单一区服网关服务器总容纳玩家数上限为50W，在实际业务上线过程中部署可容纳55W-60W人数的资源，避免部分网关服务器受DDOS攻击下线后导致整体区服资源不足

3.2.2 秘籍2：高可用架构容灾

在通过相关安全服务应对网络安全威胁时，需要同步关注链路的高可用能力，避免出现单点故障导致整体服务受损/不可用。通过高可能架构容灾设计，业务可具备更高上限的防护能力和更灵活的容灾能力。高可用架构容灾可从以下三个维度逐级设计：

1.云安全服务高可用设计

- 同一业务：设计多个业务入口域名，在域名层组成负载/主备链路，应对单一域名访问异常的场景（如域名封堵、域名解析异常）
- 同一业务：启用多个安全服务实例，在安全接入层组成负载/主备架构，应对单一安全服务实例异常的场景
- 不同业务：考虑共用多个安全服务实例资源，不同业务之间形成资源负载/主备关系，应对单一安全服务实例异常的场景

2.源站资源高可用设计

- 源站优先选用具备自动容灾能力的云网络服务（如：SLB、NAT网关），并进行相关容灾切换效果验证
- 设计多源站部署，在源站层形成负载/主备链路，通过安全接入层回源配置实现主备回源/负载均衡回源/按权重回源等效果
- 设计源站可用性检查机制，实现单一源站异常情况下自动切换源站/降低个别源站权重等效果

3.业务高可用设计

- 在业务功能逻辑进行设计，实现客户端多链路质量自动探测/多链路自动切换等效果
  - 多链路质量自动探测：指客户端根据预设或者动态下发的探测参数，对各可用链路发起探测，获取实时质量数据
  - 多链路自动切换：指客户端根据探测质量结果或者业务容错逻辑，自动切换链路以保障游戏过程中业务请求质量和可用性
- 设计业务运维工具能力，实现按需启用新链路/停用旧链路/停止链路自动切换等效果
  - 启用新链路/停止旧链路：在业务维护、故障应急等场景下对链路进行启停动作，完成业务流量优雅引流/排水
  - 停止链路自动切换：在重保场景（如：游戏活动、游戏开服）中按需关闭链路自动切换，降低切换带来的不确定性

3.2.3 秘籍3：全链路多重防护

由于安全产品技术无法百分百契合业务安全防护需求，游戏业务无法仅依靠安全服务对抗所有潜在的网络安全风险，因此需要在业务全链路中逐级设计，通过全链路多重防护满足需求。

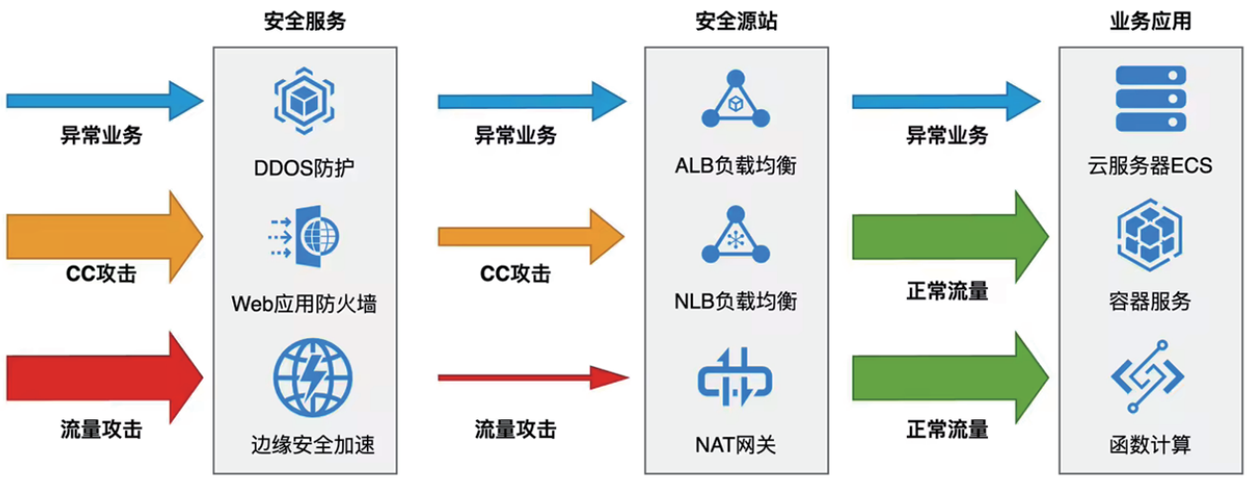

图33

1.云安全服务防护设计目标

- 完成绝大部分（约90%以上）流量攻击的防护工作
- 完成大部分（约70%以上）CC攻击的防护工作
- 部分场景下拦截特定客户端IP流量，如：拦截游戏灰产风险IP流量

2.源站资源防护设计目标

- 完成小部分（约10%）流量攻击的防护工作
- 完成小部分（约30%以下）CC攻击的防护工作

3.业务防护设计目标

- 基于其他业务特征在业务层完成防护工作

3.2.4 秘籍4：全链路可用性监控

1.服务端视角：业务与云服务可用性监控与告警

游戏服务端可用性监控分为业务维度与云服务维度，其中云服务维度可参考下文《网络安全监控告警实践建议》部分，目前支持的监控工具及指标能覆盖大部分游戏业务网络安全场景，暂不赘述。

在设计与落地游戏服务端业务维度的监控过程中，可参考以下重要原则：

- 如不同生产环境（游戏官服、游戏渠道服、游戏体验服、内部测试服）共享云服务资源/在网络层面互访互通，建议监控告警覆盖所有游戏区服，避免单一区服故障影响多个游戏区服时发现与响应滞后。

2.第三方视角：服务端可用性监控与告警

除了通过服务端业务维度、云服务维度进行监控，还可以通过第三方视角对服务端可用性进行监控，常见方案为使用拨测服务持续发起探测并发出告警。在配置拨测服务过程中，可参考以下建议：

- 合理配置拨测服务的探测频次与方式，避免探测流量过大/探测任务消耗资源过多影响游戏游戏服务端可靠性/带来预期外成本增加。
- 尽可能覆盖真实玩家所在地区、运营商、客户端版本、客户端类型，增强发现“小概率异常事件”的成功率（如：某省市运营商封禁个别游戏业务域名）。
- 可针对历史上曾发生重大网络安全事件的业务入口、服务节点做单独拨测，细化监控粒度，降低同类问题应急处置时间成本与业务成本。

3.客户端视角：客户端、中间链路及服务端可用性监控

相比于服务端视角、第三方视角，客户端视角监控技术难度及成本是相对较高的，需要综合考虑客户端类型、版本、网络环境等各种因素综合设计实现；但客户端视角监控也是最具说服力的，因为其反映了真实终端玩家访问游戏服务端的可用性和游戏体验。如游戏项目有相对充足的成本进行客户端监控，在设计与落地客户端监控的过程中，可参考以下建议：

- 优先覆盖基础技术指标（如：网络联通性、网络延迟、业务域名解析成功率...），满足基本排障需求；逐步更迭监控能力，按需覆盖有利于稳定性建设的其他指标（如：核心业务接口调用成功率、HTTP请求调用耗时分布...）。
- 兼顾备用业务链路监控，不仅能校验备用链路稳定性，在客户端具备容灾能力的情况下可同步验证容灾功能可用性。
- 合理配置客户端测试频次与方式，避免探测流量过大/探测任务消耗资源过多影响游戏客户端主要进程可靠性。
- 合理设计客户端结果缓存机制与上报链路，在客户端断网/客户端进程退出等不具备上报条件的情况下延后上报结果，便于保留问题现场探测数据。

3.3 如何做好游戏业务网络安全重保

3.3.1 Step 1：网络安全防护架构与能力梳理

和游戏业务中其他重保场景一样，在进行网络安全重保前，需要先对业务链路和能力进行梳理，以便于后续的重保实施。

1.链路架构梳理

- 主用/备用业务链路中业务流量经过的服务节点，包括但不限于：业务域名、公网IP、业务端口、云服务实例、云服务规格、基本业务逻辑。
- 主用/备用业务链路容灾决策链路及技术原理

2.链路能力梳理

- 主用业务链路及备份业务链路各节点的性能水位梳理，包括但不限于：新建/并发连接数、QPS、业务带宽、防护带宽、清洗/黑洞阈值、业务性能上限。
- 主用业务链路切换至备份业务链路的性能表现，包括但不限于：切换总耗时、各步骤耗时、业务表现、切换成功率。

3.3.2 Step 2：网络安全配置巡检及实践建议

1. DDoS原生防护配置

- **流量清洗阈值：**建议使用默认阈值，满足大部分防护场景。个别游戏业务场景**（如：依赖KCP协议进行快速C/S通信，正常业务流量中小报文占比较多导致pps偏高）**，经评估后可适当提高pps清洗阈值，避免正常流量在较低pps状态下进入清洗。配置方法：
https://help.aliyun.com/zh/anti-ddos/anti-ddos-origin/user-guide/configure-a-traffic-scrubbing-threshold
- 防护策略
  - **默认防护策略：**在新游项目上线初期，建议采用“正常”防护级别的默认防护策略，以满足大多数业务场景的需求。同时在正式上线前，推荐通过触发清洗事件来验证所选策略的有效性和准确性，确保防护措施既能有效抵御DDoS攻击，又不会对正常用户访问造成影响。默认防护策略配置参考：
https://help.aliyun.com/zh/anti-ddos/anti-ddos-origin/user-guide/set-global-mitigation-policy

- **自定义防护策略：**如部分公网资产有明确的防护需求，可启用自定义防护策略实现更为丰富的防护效果。自定义防护策略配置参考：
https://help.aliyun.com/zh/anti-ddos/anti-ddos-origin/user-guide/ip-protection-policy

2.新BGP高防/国际高防配置

- 实例规格配置
  - **功能套餐：**按需选择**标准功能/增强功能**即可。两者差异可参考：
https://help.aliyun.com/zh/anti-ddos/anti-ddos-pro-and-premium/product-overview/function-plan
  - **正常业务带宽、正常业务QPS：**根据业务用量按需配置。
  - **防护带宽：**根据历史安全事件异常流量峰值带宽进行配置，如某业务域名历史最高攻击带宽100Gbps，则防护带宽建议配置不低于100Gbps；可考虑在不同业务域名之间共用防护带宽资源，提高防护资源利用率。
  - **弹性业务带宽：建议开启**，开启后可在成本可控条件下实现较大的业务带宽突发，适用于游戏业务重大活动场景**（如：游戏周年庆活动、新游开服公测、游戏重大版本开服……）**。配置参考：
https://help.aliyun.com/zh/anti-ddos/anti-ddos-pro-and-premium/user-guide/manage-anti-ddos-pro-or-anti-ddos-premium-instances#section-hgz-se9-9i6
  - **弹性QPS：建议开启**，理由同**弹性业务带宽**，开启后可实现更高的业务吞吐。配置参考：
https://help.aliyun.com/zh/anti-ddos/anti-ddos-pro-and-premium/user-guide/manage-anti-ddos-pro-or-anti-ddos-premium-instances#section-890-i4m-yb2
  - **弹性防护带宽：建议开启**，理由同**弹性业务带宽**，开启后可实现更高的安全防护上限，应对更加复杂的安全防护场景。请注意，实例可配置的弹性防护带宽上限由防护带宽决定，其中只有300G防护带宽才能实现“全力防护”的弹性防护带宽效果。配置参考：
https://help.aliyun.com/zh/anti-ddos/anti-ddos-pro-and-premium/user-guide/manage-anti-ddos-pro-or-anti-ddos-premium-instances#section-0ti-xgt-cp3
- **TLS安全策略：建议选择支持TLS 1.0及以上版本**，以适应更加复杂的客户端TLS版本场景。如DDoS高防承载的业务为2B业务**（如：游戏社区广告RTA业务）**且客户端使用特定版本TLS协议进行握手通信，可按需调整TLS安全策略，以满足更为严格的安全防护需求。
- **流量标记：**根据业务用量按需配置。部分业务场景中（如：游戏风控业务场景，需获取客户端真实IP用于风控业务分析），流量标记可在DDoS高防服务端对七层请求流量打上特定Header，便于源站基于Header信息做下一步逻辑处理。
- **通用防护策略**
  - 基础设施DDoS防护
    - **建议启用：全局防护策略**。优先采用正常防护等级，待业务运行1-2周后根据是否存在误拦截调整防护等级。
    - **按需启用：黑白名单、区域封禁、近源流量压制、UDP反射攻击防护**。此类防护功能针对特定安全场景，暂不展开介绍。
  - 网站业务DDoS防护
    - **建议启用：AI智能防护、DDoS全局防护策略**。其中，启用AI智能防护后建议在新业务上线期间设置其模式为“预警”，使其在业务运行过程中智能学习业务流量特征，运行3天或以上再调整为“防护”；DDoS全局防护策略优先选择“正常”，运行1-2周后根据实际防护情况按需调整为“宽松”或者“严格”。
    - 按需启用：**黑白名单、区域封禁、CC安全防护**。此类防护功能针对特定安全场景，暂不展开介绍。
  - 非网站业务DDoS防护
    - **建议启用：四层AI智能防护**。启用四层AI智能防护后DDoS实例会智能学习流量业务

特征,预计耗时3天；建议选用“正常”防护等级，运行1-2周后根据实际防护情况按需调整为“宽松”或者“严格”。

- 按需启用：DDoS防护策略。此类防护功能针对特定安全场景，暂不展开介绍。

- 源站配置建议
  - 源站性能容量应大于DDoS服务性能容量，避免出现DDoS服务未满载情况下，源站由于性能容量不足导致业务受损。
  - 源站应配置白名单/ACL规则，放行来自DDoS服务的回源流量，拒绝其他公网IP访问流量，实现源站防护。
  - 如源站配置为多个域名，DDoS服务对多个源站进行转发时，会先将多个域名先解析为多个IP地址，根据IP地址进行回源负载转发，因此可能出现“某个源站域名解析出来的IP地址数目较多导致接收到的回源请求较多”，从技术上符合预期。配置多域名源站时需要予以注意源站cname解析的A记录数量对比。

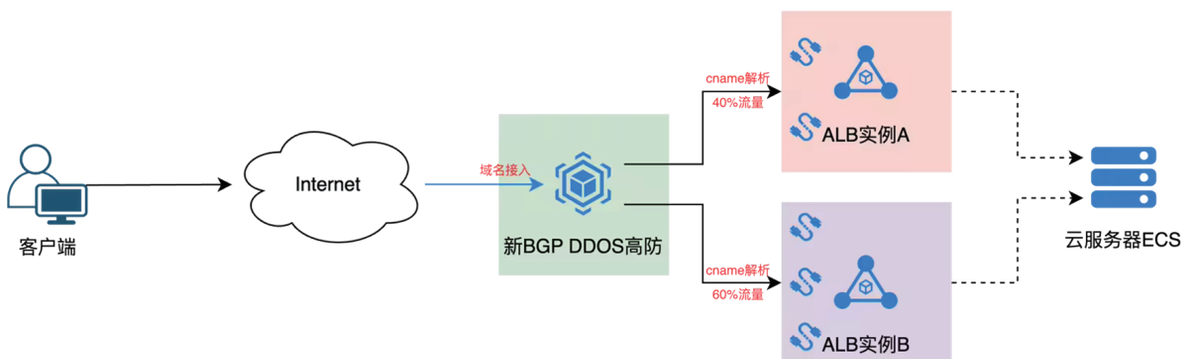

图34

- 如源站同时部署在其他无固定回源IP地址的代理服务（如：CDN）后端，建议分多源站部署：其中源站A部署在DDoS服务后端，通过白名单放行DDoS回源流量；源站B部署在CDN服务后端，不限制访问来源IP地址。在使用此类主备链路架构时，请考虑在源站通过流量标记或者Host进行检查，避免出现非预期的域名访问。

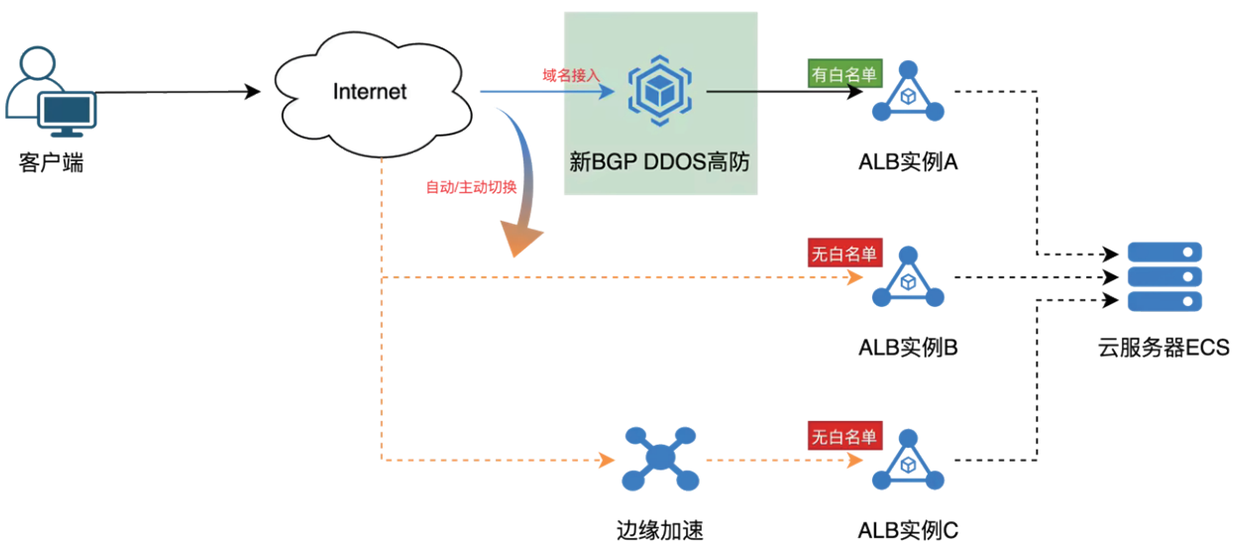

图35

3.3.3 Step 3：网络安全监控告警实践建议

安全类产品

DDoS原生防护监控和告警建议

时序指标

表3

云产品类型	CMS指标名称	CMS 指标含义	CMS 指标单位	是否 接入监控	是否 配置告警
DDoS原生防护	PacketRateIn	入向数据包速率	pps	是	是
DDoS原生防护	TrafficRateIn	入向带宽	bit/s	是	是

系统事件

表4

云产品类型	CMS事件名称	CMS 指标含义	CMS 事件等级	是否 配置警告
DDoS原生防护	ddosbgp_event_blackhole	黑洞	CRITICAL	是
DDoS原生防护	ddosbgp_event_clean	清洗	CRITICAL	是

新BGP高防/国际高防监控和告警建议

时序指标

表5

云产品类型	CMS指标名称	CMS 指标含义	CMS 指标单位	是否 接入监控	是否 配置告警
新BGP高防/国际高防	Active_connection	活跃连接数	count	是	是
新BGP高防/国际高防	AttackTraffic	高防IP攻击流量	bit/s	是	是
新BGP高防/国际高防	Back_Traffic	高防IP回源流量	bit/s	是	是
新BGP高防/国际高防	In_Traffic	高防IP入流量	bit/s	是	是
新BGP高防/国际高防	Inactive_connection	非活跃连接数	count	是	是
新BGP高防/国际高防	New_connection	新建连接数	count	是	是
新BGP高防/国际高防	Out_Traffic	高防IP出流量	bit/s	是	是
新BGP高防/国际高防	qps	QPS	count/s	是	是
新BGP高防/国际高防	qps_ratio_down	QPS环比下降率	%	是	是
新BGP高防/国际高防	qps_ratio_up	QPS环比增长率	%	是	是
新BGP高防/国际高防	resp2xx	2XX状态码	count	是	否
新BGP高防/国际高防	resp3xx	3XX状态码	count	是	否
新BGP高防/国际高防	resp4xx	4XX状态码	count	是	是
新BGP高防/国际高防	resp5xx	5XX状态码	count	是	是
新BGP高防/国际高防	upstream_resp4xx	4XX回源状态码	count	是	是
新BGP高防/国际高防	upstream_resp5xx	5XX回源状态码	count	是	是

系统事件

表6

云产品类型	CMS指标名称	CMS 事件含义	CMS 事件等级	是否 配置告警
新BGP高防	ddoscoo_event_blackhole_add	黑洞进行中	CRITICAL	是
新BGP高防	ddoscoo_event_blackhole_end	黑洞解除	CRITICAL	是
新BGP高防	ddoscoo_event_cc4_add	4层cc攻击进行中	CRITICAL	是
新BGP高防	ddoscoo_event_cc4_end	4层cc攻击结束	CRITICAL	是
新BGP高防	ddoscoo_event_cc7_add	7层cc攻击进行中	CRITICAL	是
新BGP高防	ddoscoo_event_cc7_end	7层cc攻击结束	CRITICAL	是
新BGP高防	ddoscoo_event_defense_add	清洗进行中	CRITICAL	是
新BGP高防	ddoscoo_event_defense_end	清洗解除	CRITICAL	是
国际高防	ddosdip_event_blackhole_add	黑洞进行中	CRITICAL	是
国际高防	ddosdip_event_blackhole_end	黑洞解除	CRITICAL	是
国际高防	ddosdip_event_cc4_add	4层cc攻击进行中	CRITICAL	是
国际高防	ddosdip_event_cc4_end	4层cc攻击结束	CRITICAL	是
国际高防	ddosdip_event_cc7_add	7层cc攻击进行中	CRITICAL	是
国际高防	ddosdip_event_cc7_end	7层cc攻击结束	CRITICAL	是
国际高防	ddosdip_event_defense_add	清洗进行中	CRITICAL	是
国际高防	ddosdip_event_defense_end	清洗解除	CRITICAL	是

Web应用防火墙监控和告警建议

时序指标

表7

云产品类型	CMS指标名称	CMS 指标含义	CMS 指标单位	是否 接入监控	是否 配置告警
Web应用防火墙	4XX_ratio	4XX占比	%	是	是
Web应用防火墙	5XX_ratio	5XX占比	%	是	是
Web应用防火墙	acl_blocks_5m	访问控制拦截量(5m)	count	是	是
Web应用防火墙	cc_blocks_5m	CC防护拦截量(5m)	count	是	是
Web应用防火墙	qps	QPS	countS	是	是
Web应用防火墙	qps_ratio	QPS环比增长率	%	是	是
Web应用防火墙	qps_ratio_down	QPS环比下降率	%	是	是

云产品类型	CMS指标名称	CMS 指标含义	CMS 指标单位	是否 接入监控	是否 配置告警
Web应用防火墙	waf_blocks_5m	4XX占比	count	是	是
Web应用防火墙	waf_qps	5XX占比	countS	是	是
Web应用防火墙	4XX_ratio-wafv3	访问控制拦截量(5m)	%	是	是
Web应用防火墙	5XX_ratio-wafv3	CC防护拦截量(5m)	%	是	是
Web应用防火墙	acl_blocks_5m-wafv3	QPS	count	是	是
Web应用防火墙	cc_blocks_5m-wafv3	QPS环比增长率	count	是	是
Web应用防火墙	qps-wafv3	QPS环比下降率	countS	是	是
Web应用防火墙	qps_ratio-wafv3	访问控制拦截量(5m)	%	是	是
Web应用防火墙	qps_ratio_down-wafv3	CC防护拦截量(5m)	%	是	是
Web应用防火墙	waf_blocks_5m-wafv3	QPS	count	是	是
Web应用防火墙	waf_qps-wafv3	QPS环比增长率	countS	是	是
Web应用防火墙	waf_qps-wafv3-max	QPS环比下降率	countS	是	是

系统事件

表8

云产品类型	CMS指标名称	CMS 事件含义	CMS 事件等级	是否 配置告警
Web应用防火墙	waf_event_aclattack	访问控制事件	CRITICAL	是
Web应用防火墙	waf_event_bandwidth_exceed	带宽超限	CRITICAL	是
Web应用防火墙	waf_event_ccattack	CC攻击事件	CRITICAL	是
Web应用防火墙	waf_event_qps_exceed	QPS超限	CRITICAL	是
Web应用防火墙	waf_event_webattack	Web攻击事件	CRITICAL	是
Web应用防火墙	waf_event_webscan	防扫描事件	CRITICAL	是
Web应用防火墙	wafv3_event_aclattack	访问控制事件V3	CRITICAL	是
Web应用防火墙	wafv3_event_apisec	API安全事件V3	CRITICAL	是
Web应用防火墙	wafv3_event_ccattack	CC攻击事件V3	CRITICAL	是
Web应用防火墙	wafv3_event_webattack	Web攻击事件V3	CRITICAL	是
Web应用防火墙	wafv3_event_webscan	防扫描事件V3	CRITICAL	是
Web应用防火墙	xray_wafv3_event_cost_protection	计费保护触发事件V3	CRITICAL	是
Web应用防火墙	xray_wafv3_event_log_exceed	日志容量超用事件V3	CRITICAL	是

云产品类型	CMS指标名称	CMS 事件含义	CMS 事件等级	是否 配置告警
Web应用防火墙	xray_wafv3_event_qps_exceed	QPS超用事件V3	CRITICAL	是
Web应用防火墙	xray_wafv3_evnet_migrate_lost_days	WAFv3迁移到期前告警	CRITICAL	是

非安全类产品

在已经使用安全类产品作为业务链路的安全接入层的前提下，到达源站服务的威胁流量已经降到一个较低量级，因此非安全类产品的监控与告警主要以水位监控和异常发现为主。

弹性公网IP/共享带宽监控和告警建议

时序指标

表9

云产品类型	CMS指标名称	CMS 指标含义	CMS 指标单位	是否 接入监控	是否 配置告警
弹性公网IP	in_ratelimit_drop_speed	入方向限速丢包速率	pps	是	是
弹性公网IP	net_in.rate_percentage	流入带宽使用率	%	是	是
弹性公网IP	net_out.rate_percentage	流出带宽使用率	%	是	是
弹性公网IP	net_rx.rate	流入带宽	bit/s	是	否
弹性公网IP	net_tx.rate	流出带宽	bit/s	是	否
弹性公网IP	out_ratelimit_drop_speed	出方向限速丢包速率	pps	是	是
共享带宽	in_bandwidth_utilization	流入带宽使用率	%	是	是
共享带宽	in_ratelimit_drop_pps	入方向限速丢包速率	pps	是	是
共享带宽	net_rx.rate	流入带宽	bit/s	是	否
共享带宽	net_tx.rate	流出带宽	bit/s	是	否
共享带宽	out_bandwidth_utilization	流出带宽使用率	%	是	是
共享带宽	out_ratelimit_drop_pps	出方向限速丢包速率	pps	是	是

系统事件：

弹性公网IP和共享带宽暂无相关系统事件

增强型NAT网关监控和告警建议

时序指标

表10

云产品类型	CMS指标名称	CMS指标含义	CMS指标单位	是否接入监控	是否配置告警
增强型NAT网关	BWRateInFromInside	从VPC来流量速率	bit/s	是	否
增强型NAT网关	BWRateInFromOutside	从公网来流量速率	bit/s	是	否
增强型NAT网关	BWRateOutToInside	入VPC流量速率	bit/s	是	否
增强型NAT网关	BWRateOutToOutside	入公网流量速率	bit/s	是	否
增强型NAT网关	DropTotalBps	报文丢弃总带宽	bit/s	是	是
增强型NAT网关	DropTotalPps	报文丢弃总速率	count/s	是	是
增强型NAT网关	SessionActiveConnectionWaterLever	并发连接水位	%	是	是
增强型NAT网关	SessionLimitDropConnection	并发丢弃连接速率	count/s	是	是
增强型NAT网关	SessionNewConnectionWaterLever	新建连接水位	%	是	是
增强型NAT网关	SessionNewLimitDropConnection	新建丢弃连接速率	count/s	是	是

系统事件：

增强型NAT网关暂无相关系统事件

CLB经典负载均衡监控和告警建议

时序指标

表11

云产品类型	CMS指标名称	CMS指标含义	CMS指标单位	是否接入监控	是否配置告警
CLB经典负载均衡	InstanceDropTrafficRX	实例每秒丢失入bit数	bit/s	是	是
CLB经典负载均衡	InstanceDropTrafficTX	实例每秒丢失出bit数	bit/s	是	是
CLB经典负载均衡	InstanceMaxConnectionUtilization	实例最大连接数使用率	%	是	是
CLB经典负载均衡	InstanceNewConnectionUtilization	实例新建连接数使用率	%	是	是
CLB经典负载均衡	InstanceQpsUtilization	七层实例QPS使用率	%	是	是
CLB经典负载均衡	InstanceStatusCode4xx	七层实例每秒状态码4XX数量	count/s	是	是
CLB经典负载均衡	InstanceStatusCode5xx	七层实例每秒状态码5XX数量	count/s	是	是
CLB经典负载均衡	InstanceTrafficRXUtilization	实例网络流入带宽使用率	%	是	是
CLB经典负载均衡	InstanceTrafficTXUtilization	实例网络流出带宽使用率	%	是	是
CLB经典负载均衡	UnhealthyServerCount	健康检查后端异常ECS实例个数	count	是	是

系统事件：

暂无，CLB目前支持的系统事件与网络安全防护不直接相关

NLB网络型负载均衡监控和告警建议

时序指标

表12

云产品类型	CMS指标名称	CMS指标含义	CMS指标单位	是否接入监控	是否配置告警
网络型负载均衡	InstanceActiveConnection	实例每秒活跃连接数	count	是	否
网络型负载均衡	InstanceDropConnection	实例每秒丢弃连接数	count/s	是	是
网络型负载均衡	InstanceDropPacketRX	实例每秒丢弃入包数	count/s	是	是
网络型负载均衡	InstanceDropPacketTX	实例每秒丢弃出包数	count/s	是	是
网络型负载均衡	InstanceInactiveConnection	实例每秒非活跃连接数	count/s	是	否
网络型负载均衡	InstanceMaxConnection	实例每秒最大并发连接数	count/s	是	是
网络型负载均衡	InstanceNewConnection	实例每秒新建连接数	count/s	是	否
网络型负载均衡	InstancePacketRX	实例每秒入包数	count/s	是	否
网络型负载均衡	InstancePacketTX	实例每秒出包数	count/s	是	否
网络型负载均衡	InstanceTrafficRX	实例每秒入bit数	bit/s	是	否
网络型负载均衡	InstanceTrafficTX	实例每秒出bit数	bit/s	是	否
网络型负载均衡	InstanceUnhealthyServerCount	实例健康检查后端异常ECS实例个数	count	是	是
网络型负载均衡	NlbInstanceHeathyServerCount	Nlb实例健康后端个数	count	是	否

系统事件：

增强型NAT网关暂无相关系统事件

ALB应用型负载均衡监控和告警建议

时序指标

表13

云产品类型	CMS指标名称	CMS指标含义	CMS指标单位	是否接入监控	是否配置告警
应用型负载均衡	LoadBalancerActiveConnection	负载均衡实例活跃连接数	count	是	否
应用型负载均衡	LoadBalancerClientTLSNegotiationError	负载均衡实例每秒TLS握手失败连接数	count/s	是	是
应用型负载均衡	LoadBalancerHTTPCode4XX	负载均衡实例每秒4XX个数	count/s	是	是
应用型负载均衡	LoadBalancerHTTPCode5XX	负载均衡实例每秒5XX个数	count/s	是	是
应用型负载均衡	LoadBalancerHTTPCodeUpstream4XX	负载均衡实例后端每秒4XX个数	count/s	是	是
应用型负载均衡	LoadBalancerHTTPCodeUpstream5XX	实例每秒最大并发连接数	count/s	是	是
应用型负载均衡	LoadBalancerHealthyHostCount	负载均衡实例健康的服务器数	count	是	是
应用型负载均衡	LoadBalancerInBits	负载均衡实例入带宽	bit/s	是	否

云产品类型	CMS指标名称	CMS 指标含义	CMS 指标单位	是否 接入监控	是否 配置告警
应用型负载均衡	LoadBalancerInactiveConnection	负载均衡实例 非活跃连接数	count	是	否
应用型负载均衡	LoadBalancerMaxConnection	负载均衡实例 每秒最大并发连接数	count/s	是	否
应用型负载均衡	LoadBalancerNewConnection	负载均衡实例 每秒新建连接数	count/s	是	是
应用型负载均衡	LoadBalancerOutBits	负载均衡实例出带宽	bit/s	是	是
应用型负载均衡	LoadBalancerQPS	负载均衡实例 每秒请求数	count/s	是	否
应用型负载均衡	LoadBalancerRejectedConnection	负载均衡实例 每秒丢弃连接数	count/s	是	是
应用型负载均衡	LoadBalancerUnHealthyHostCount	负载均衡实例 不健康的服务器数	count	是	是
应用型负载均衡	LoadBalancerUpstreamTLSNegotiationError	负载均衡实例后端 每秒TLS握手失败数	count/s	是	是

系统事件：

应用型负载均衡暂无相关系统事件

3.3.4 Step 4：云服务维度/业务维度应急预案及演练

安全类产品常见应急预案参考

表14

涉及产品类型	应急场景	业务场景	应急预案	预期效果	预案影响	回退方案/ 后续动作
基础DDoS防护	异常公网流量带宽超出防护带宽上限，公网资产进入黑洞状态	游戏对外公网资产被恶意攻击	切换至DDoS原生防护，获得更高防护能力上限	切换至DDoS原生防护提供更高黑洞阈值防护，公网资产停止黑洞，业务逐步恢复正常	切换至DDoS原生防护过程中业务公网流量不受影响	评估保持使用DDoS原生防护的必要性
DDoS原生防护	异常公网流量带宽超出防护带宽上限，公网资产进入黑洞状态	游戏网关服被恶意攻击	登录DDoS控制台查看攻击是否停止，如停止可手动解除黑洞状态 临时升配DDoS原生防护的防护带宽 如条件允许，切换公网资产或更换公网IP，需同步考虑更换IP的业务成本	DDoS原生防护提供更高黑洞阈值防护，公网资产停止黑洞，业务逐步恢复正常 更换公网IP后流量攻击停止，业务恢复正常；也有可能持续攻击，业务再次受损	DDoS原生防护升配过程中业务公网流量不受影响	评估保持使用当前防护规格 DDoS原生防护的必要性 如发现攻击规模较大且频发触发黑洞，短期内配置近源封禁策略，拦截海外流量；长期考虑使用高防EIP或DDoS高防，获得Tbps级别防护能力
DDoS原生防护	异常公网流量未达到原生防护清洗阈值（如小带宽高pps的cc攻击），公网业务受损	游戏网关服或登录服被恶意攻击	根据正常业务流量特征，调整原生防护清洗阈值 如条件允许，切换公网资产或更换公网 IP，需同步考虑更换 IP 的业务成本	DDoS 原生防护调整清洗阈值后成功清洗异常流量，业务逐步恢复正常 更换公网IP 后流量攻击停止，业务恢复正常；也有可能持续攻击，业务再次受损	切换 DDoS 原生防护清洗阈值过程中，业务正常公网流量无感知	保持使用新的清洗阈值配置 在业务低峰期或合适时间点回切清洗阈值配置

涉及产品类型	应急场景	业务场景	应急预案	预期效果	预案影响	回退方案/后续动作
新 BGP高防/国际高防	异常公网流量带宽超出防护带宽上限，公网资产进入黑洞状态	游戏网关服、登录服、客户端 SDK 接口被恶意攻击	登录DDoS控制台查看攻击是否停止，如停止可手动解除黑洞状态 如攻击持续，考虑升配高防实例防护带宽，启用弹性防护带宽，获得更高防护阈值 切换业务流量至备份安全链路 DDoS攻击期间，不建议切换业务流量绕行高防实例，避免影响加剧	升配高防实例后成功清洗异常公网流量，业务逐步恢复正常 切换链路后异常公网流量持续，但在安全防护阈值内，业务逐步恢复正常；也有可能攻击持续，且超出当前链路安全防护阈值，业务再次受损	如通过调整域名DNS配置切换至备份链路，可能出现DNS域名解析缓存导致切流过慢，业务持续受损，建议同步调整域名记录TTL	保持使用切换后的安全防护链路，在合适时间点将业务流量回切主用防护链路或降配高防实例，不建议长期暴露源站 如发现攻击规模较大且频发触发黑洞，短期内配置近源封禁策略，拦截海外异常流量
新BGP高防/国际高防/Web应用防火墙	正常公网流量被误拦截/疑似被中间节点拦截，影响正常业务	大型游戏活动导致游戏网关服、登录服、客户端SDK等接口流量突增 游戏业务逻辑调整，导致游戏网关服、登录服、客户端SDK等接口流量特征变化	基于正常流量特征，按需在高防/WAF新增流量白名单，放行部分可信流量 如具备业务日志分析能力，谨慎调整高防/WAF实例防护模式，改为“观察”/“宽松”，降低误拦截概率	调整高防/WAF实例防护模式后误拦截情况明显好转/完全恢复，业务逐步恢复正常 切换业务流量绕行高防/WAF后，有极低概率仍存在被拦截，需结合全链路分析进一步判断	调整高防/WAF实例防护模式可能导致其他异常流量被放行，调整后需要持续关注业务表现	联合阿里云工程师进行正常业务流量分析，获取更多技术建议
新BGP高防/国际高防/Web应用防火墙	异常公网流量未按被安全服务拦截，透过安全防护进入源站	攻击者伪造游戏网关服、登录服、客户端SDK等接口正常流量进行安全入侵行为；或红蓝对抗/安全扫描活动中流量绕开安全检测达到源站	基于异常流量特征，按需在高防/WAF新增安全防护策略，近源侧完成异常流量拦截/流量限速 如具备业务日志分析能力，谨慎调整高防/WAF实例防护模式，改为“观察”/“宽松”，降低误拦截概率	调整高防/WAF实例防护模式或新增安全防护策略后异常流量拦截能力明显好转/完全拦截，业务逐步恢复正常 调整高防/WAF实例防护模式或新增安全防护策略后发现正常流量误拦截，或异常流量仍未被拦截/拦截效果未达预期，建议联合阿里云工程师进一步分析	调整高防/WAF实例防护模式或新增安全防护策略可能导致误拦截/其他衍生问题，调整后需要持续关注业务表现	检查安全服务是否正确开启防护策略、是否有配置了加白策略导致异常流量被放行 如排查定位为大流量cc攻击，可选择在DDoS高防开启AI防护，结合防CC策略缓解问题 联合阿里云工程师进行技术分析讨论，获取更多技术建议

非安全类产品常见应急预案参考

表15

涉及产品类型	应急场景	业务场景	应急预案	预期效果	预案影响	回退方案/后续动作
弹性公网IP/共享带宽	公网流量带宽超出带宽上限出现丢包，业务流量受损	大型游戏活动导致游戏网关服、登录服、客户端SDK等接口流量突增 游戏业务逻辑调整导致相同业务热度状态下带来公网流量带宽增加	临时提升业务带宽上限	提升后的业务带宽满足业务需求，业务逐步恢复正常 提升后的业务带宽未能满足业务需求，业务持续影响，需考虑横向扩容带宽能力/业务分流等技术方案	原地提升业务公网带宽过程平滑 如需要更换EIP绑定的共享带宽，更换过程中可能由于短暂的公网带宽过低导致业务影响加剧，需提前做好相关应对	保持使用提升后的公网带宽 在业务低峰期或合适时间点调整EIP绑定的共享带宽，结合性能与成本重新分配共享带宽资源 如流量突增为预期外表现，建议事后进一步分析原因
增强型NAT网关	公网流量带宽/新建连接数/并发连接数超出性能上限出现丢包，业务流量受损	大型游戏活动导致游戏网关服、登录服、客户端SDK等接口流量突增 游戏业务逻辑调整导致相同业务热度状态下带来公网流量带宽增加 游戏业务机器由于代码缺陷/安全因素导致公网访问流量突增	切换业务流量至备份链路，尽可能降低业务影响面 如流量突增为预期外表现，通过NIS/云防火墙的公网流量观察能力，追溯突增流量来源 如NAT网关弹性扩容逻辑与效率未能完全满足业务流量增长需求，升级阿里云工程师做进一步应急	自动扩容/人工干预后的NAT网关性能满足业务需求，业务逐步恢复正常 自动扩容/人工干预后的NAT网关性未能满足业务需求，业务影响持续，建议联合阿里云工程师进一步分析	NAT网关原地扩容资源过程平滑 切换业务流量至备份链路可能造成短暂影响，视技术方案而定	评估部署多NAT网关的可行性与必要性 如流量突增为预期外表现，建议事后进一步分析原因
负载均衡CLB/NLB/ALB	公网流量带宽/新建连接数/并发连接数/QPS超出性能上限出现丢包，业务流量受损	大型游戏活动导致游戏网关服、登录服、客户端SDK等接口公网用量突增 游戏业务逻辑调整导致相同业务热度状态下带来公网用量增加 源站SLB信息泄露，部分流量绕行安全防护层直达源站SLB	如源站为CLB，横向扩容CLB资源，提升源站处理性能上限 如源站为NLB/ALB，关注弹性扩容能力效率与上限，并准备横向扩容NLB/ALB资源缓解资源压力 如为源站SLB信息泄露，更换源站SLB并完善SLB访问白名单规则，只放行预期内流量	自动扩容/人工干预后的SLB满足业务需求，业务逐步恢复正常 自动扩容/人工干预后的SLB未能满足业务需求，业务影响持续，建议联合阿里云工程师进一步分析	横向扩容SLB需要一定时间完成原SLB流量迁移，具体效率取决于业务规模及负载均衡策略，期间业务可能持续受损后恢复正常 NLB/ALB自动扩容过程平滑 更换源站SLB过程中可能影响加剧，取决于具体技术方案能力	评估部署扩容SLB资源/替换CLB为NLB和ALB的可行性与必要性 如为源站信息泄露，建议事后进行相关安全溯源，避免再次发生/情况加剧 如流量突增为预期外表现，建议事后进一步分析原因
负载均衡CLB/NLB/ALB	公网异常流量透过安全接入层进入负载均衡SLB，业务受损	攻击者伪造游戏网关服、登录服、客户端SDK等接口正常流量进行安全入侵行为；或红蓝对抗/安全扫描活动中流量绕开安全检测达到源站	负载均衡基于异常流量特征新增转发配置/限速配置，尽可能调整拦截异常流量	调整后的负载均衡全部拦截/部分拦截异常流量，业务逐步恢复正常 调整后的负载均衡拦截效果为达到业务最低预期，业务持续影响，建议联合阿里云工程师进一步分析	调整负载均衡转发配置/限速配置过程平滑	评估优化负载均衡转发配置/限速配置的可行性与必要性 如异常流量无法在安全接入层及负载均衡进行拦截处理，建议联合阿里云工程师作进一步安全链路优化评估

游戏业务应急预案参考

表16

业务类型	应急场景	应急预案	技术要点
游戏网关服务	游戏网关服受DDoS影响进入黑洞状态，单次攻击规模大/累计攻击频次高 游戏网关公网IP被中间网络节点拦截，部分网关服务受损	下线/更换网关服务公网IP-->降低网络安全异常事件影响面 扩容网关服务资源（公网IP及关联资源）-->提升游戏区服整体水位	需要业务逻辑、运维工具支持网关资源下线/更换/扩容能力
游戏注册、登录、支付等服务接口	服务受DDoS影响进入黑洞状态，单次攻击规模大/累计攻击频次高	主动切换所有或部分业务流量至备份安全链路-->降低原安全流量业务压力 如安全接入层为多实例部署，且DDoS流量集中在单实例上，主动下线受攻击的实例-->绕行流量黑洞节点，实现业务快速止血 主动切换所有业务流量绕行安全防护层，直接访问源站-->绕行流量黑洞节点，实现业务快速止血	需要业务逻辑、运维工具支持接口服务切换链路能力 需要提前设计、测试备份链路可行性及切换效果
游戏注册、登录、支付等服务接口	业务域名出现DNS解析超时、DNS解析劫持等事件，影响业务域名DNS解析成功率	主动切换所有或部分业务流量至备份业务域名-->降低线上生产影响面 联系DNS解析服务提供商做进一步技术排查	需要业务逻辑、运维工具支持接口服务切换链路能力 需要提前设计、测试备份链路可行性及切换效果 如有条件，主备业务域名使用不同DNS服务提供商

第四章 游戏下载：加速玩家体验的极速通道

在游戏行业，游戏开服往往会引发流量的突发，这不仅考验着系统的承载能力，也对运维团队提出了更高的要求。本文将从整体的高可用架构、安全防护、配置巡检、性能优化、运维监控、容灾预案和活动准备等方面，详细探讨如何应对游戏下载带来的流量突发，以确保系统的稳定运行和用户体验的流畅。希望本文的内容和注意事项能够为大家提供有价值的参考，更好地应对游戏下载带来的流量突发，确保系统的稳定运行和用户的良好体验。

4.1 高可用架构

4.1.1 典型的多CDN厂商+主备源站架构

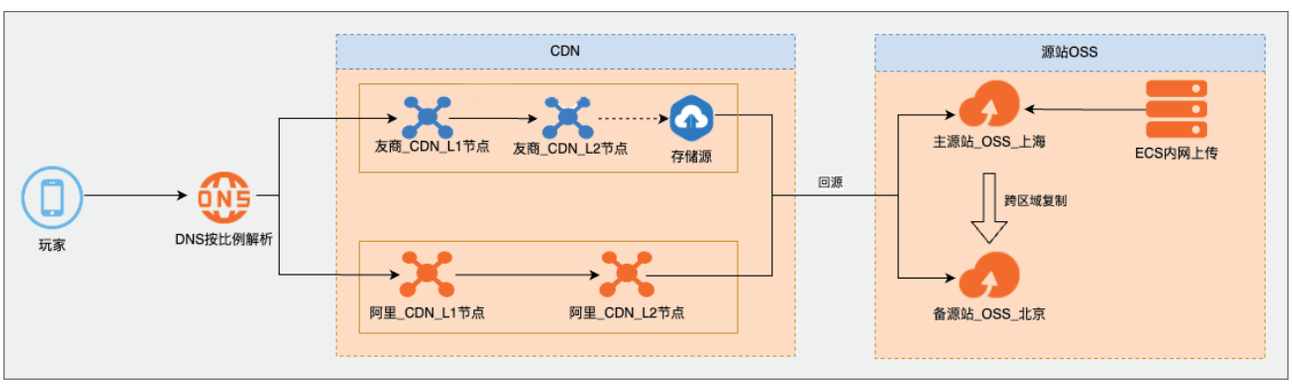

图36

4.1.2 源站OSS高可用

1. 地域选择

对于面向全球的游戏来说，针对不同区服提供不同的下载包体，因此通常是国服下载将源站OSS部署在国内，海外下载将源站OSS部署在海外，避免海外CDN从国内拉取数据产生的跨境网络质量抖动。建议选择综合能力强大的机房作为主站点，具备更高的公网带宽弹性能力，例如中国大陆地区上海、杭州、北京等，亚太地区新加坡等(具体需结合实际的业务需求结合产品资源情况等综合评估)。

2. 同城冗余

OSS采用多可用区（AZ）内的数据冗余存储机制，将用户的数据冗余存储在同一地域（Region）的多个可用区。当某个可用区不可用时，仍然能够保障数据的正常访问。如果没有开启同城冗余存储，会导致当出现某个机房不可用时，OSS服务无法提供一致性服务，影响数据恢复目标。目前OSS绝大部分的Region均已支持同城冗余，建议在创建Bucket时选择开启同城冗余，存量单AZ的Bucket也支持控制台无损切换成同城冗余。

3. 跨地域灾备

同城冗余解决的是AZ级别的容灾，可能出现无法覆盖极端场景的情况，如果对数据安全性和可用性有极高的要求，考虑地域级别的灾备，以备发生特大灾难（如地震、海啸等）导致一个OSS数据中心损毁时，还能启用另一个OSS数据中心的备份数据。OSS跨区域复制功能满足Bucket跨区域容灾的需求，目标Bucket中的Object是源Bucket中Object的精确副本，它们具有相同的Object名、版本信息、元数据以及内容，例如创建时间、拥有者、用户定义的元数据、Object ACL、Object内容等。部分地域还支持开通RTC，开通RTC后OSS会在10分钟内复制99.99%的对象，同时进行数据复制的准实时监控，具体参考官方文档。

4.1.3 CDN高可用方案

1. 多CDN供应商

针对游戏热门，通过预约玩家等运营数据预估可能有较大带宽的突发下载，为确认有更充足的资源供应及更高的服务可用性、容错能力，可能会选择多个CDN供应商，不同区域运营商均按一定调度策略和比例解析到各云厂商CDN的CNAME。当A厂商的CDN分发节点出现异常，或者该终端到A云厂商的分发节点之间链路异常，依赖客户端重试能力自动快速重试到B厂商以及手动更改解析权重的方式来实现切量，实现异常问题快速逃逸，保证服务的连续性。

2. CDN主备源站

前文已经介绍通过创建跨区域灾备的Bucket作为备份源站，可以在主源站异常时将CDN源站切换至备份源站。CDN支持添加主备源站OSS，主备源站通过OSS跨区域复制确保数据一致性。CDN正常回源到主源，当主源异常时(如回源建联失败或源站异常导致CDN 5XX)，依赖CDN的健康检查机制和容灾切换机制，自动重试切换到备源恢复。

3. CDN源站切换策略

- TCP连接异常：如果CDN节点与源站IP地址之间连续两次出现TCP连接不可用（建连失败或连接超时），CDN会从可用源站地址列表中剔除该源站IP地址，并将该IP地址加入dead table中，这样后续的回源请求就不会去访问这个源站IP地址；此后CDN节点会每隔5秒使用TCP建连去探测一次该源站IP地址，如果建连成功，则将该源站IP地址恢复到可用源站地址列表中。

- TCP连接正常：如果CDN节点与源站IP地址之间TCP连接正常，但收到源站响应的重试状态码（例如：5xx），此时虽然会触发重试的逻辑，但该源站IP地址仍然还在可用源站地址列表中，下次访问还会按权重去请求该源站（即TCP四层连接正常的情况下，七层HTTP请求异常不会主动屏蔽源站IP地址，如果需要在七层HTTP请求异常的情况下主动屏蔽源站IP地址，则需要联系阿里云技术支持申请特殊配置）。具体参见源站的健康检查策略官网文档介绍。

4.1.4 业务架构高可用

1. 域名拆分

对于包体较大且热度较高的游戏，在游戏开放下载后会造成较高的突发带宽，可能会对CDN服务商产生较大的冲击，对CDN节点带宽、调度系统能力等均有较高的要求。虽然CDN的节点很多、分布很广，但每个节点容量、水位和性能有限制并且各个节点不会完全一样。在DNS解析的场景下，因为DNS协议本身就有传输字节大小限制，通常一次解析最多只能返回13个IP，而国内各个CDN厂商均没有使用Anycast，在非常大流量的情况下，可能会要求单IP能承接的带宽很大，有可能超出单节点的承接大小，这样会触发单IP大流量的切换，而其他顶上的IP在承接后也会切换，会产生系统性风险。 为什么建议拆分，有考虑到以下两点收益:

- 流量块减小，降低单节点因业务突发等跑满的概率，解决单调度域流量块太大，无法通过给域名切换调度域来分流问题，对质量稳定也是有提升的。技术建议的方案通常是拆分域名，让不同的域名可以用不同的资源池，这样可以扩大资源池的容量。
- 同时，多域名架构下，如遇到某域名因特殊原因被运营商给强制封禁等极端情况，可以切换业务到另外域名，规避单点域名问题导致业务中断。

综上，一般来说，如果单域名的峰值带宽会超20T(不同CDN服务商的瓶颈可能有差异)，建议拆分域名，尽量控制单域名的峰值带宽在10T以下。

2. HTTPDNS

部分Local DNS供应商为了降低运营成本，会将解析请求转发给其他供应商的Local DNS节点，产生域名劫持问题，可能对域名解析的精准性带来严重影响。使用HTTPDNS服务，域名解析请求直接发送至HTTPDNS服务器，绕过运营商Local DNS，避免域名劫持问题。采用HTTPDNS这项技术使得用户终端可以绕开运营商的Local DNS，直接采用HTTP协议去访问调度系统，请求所需要访问的域名的最优接入节点，这样可以避免DNS劫持所带来的业务安全问题(需要客户端兼容支持)。

3. 客户端数据一致性校验

在网络通信中，数据的传输可能会受到干扰、丢失或错误等问题。如客户端下载到异常的游戏包体数据将导致无法正常安装或更新游戏，从而无法正常进行游戏。客户端数据一致性校验可以帮助检测数据是否被篡改或损坏，以确保数据的完整性。客户端通过校验数据的哈希值或签名等方式，可以验证数据在传输或存储过程中是否发生了变化，防止数据被恶意篡改或意外损坏。

4.2 安全防护

4.2.1 源站OSS安全防护

1. 隐藏源站

部分客户在配置中，可能会习惯性把OSS的默认公网域名(如：bucketname.oss-cn-shanghai.aliyuncs.com)或传输加速域名(如：bucketname.oss-accelerate.aliyuncs.com)配置成CDN的回源Host，这样配置会带来两个问题：

- 正常情况下OSS域名已被隐藏在CDN之后，但如果是httpcode 400、403、404等报错的场景，XML报错信息的HostId字段会暴露OSS域名，产生OSS域名被攻击的风险，如非法攻击者拿到OSS地址，则可以绕过CDN直接攻击OSS从而导致业务受损。
- 为了持续提升阿里云对象存储服务（OSS）的安全性和合规性管理，访问MIMETYPE为application/vnd.android.package-archive或application/iphone（即APK和IPA文件）的文件，请求将被OSS阻断，并返回错误码ApkDownloadForbidden，HTTP状态码400，详情请参见公告。

针对这种情况，需要绑定自定义域名至Bucket默认域名，然后CDN配置回源Host为绑定Bucket的自定义域名，这样可以解决ApkDownloadForbidden问题，同时在OSS报错场景下，报错XML信息的HostId为绑定的自定义域名(绑定自定义域名可以直接使用CDN域名，也可以使用其他域名，但是域名不能解析到Bucket域名，否则也会产生泄露)，而非Bucket默认域名，可以解决Bucket地址泄露问题。

2. 源站OSS权限配置

为了提升OSS的安全性，我们建议将Bucket设置为私有权限（可以起到访问鉴权的作用，避免非授权的请求盗刷流量），阿里CDN侧可以开启"OSS私有Bucket回源"功能，开启了私有Bucket回源功能后，阿里CDN节点将会在回源请求中添加名为 “Authorization” 的Header，其值为OSS私有Bucket鉴权签名信息。考虑到客户可能有同时接入友商CDN回源阿里OSS，友商CDN可能不支持OSS鉴权，则需要降级OSS权限至公共读，但禁止将Bucket设置为公共读写、或为匿名账号授予任何写入权限。

4.2.2 CDN安全防护

1. HTTPS加速

HTTPS安全传输，有效防止HTTP明文传输中的窃听、篡改、冒充和劫持风险。数据传输过程中对数据进行完整性校验，防止DNS或内容遭第三方劫持、篡改等中间人攻击（MITM）隐患，建议配置证书开启HTTPS，推荐开启安全级别更高的TLSv1.3版本。(确定使用哪个TLS协议取决于服务器和客户端之间的TLS协商过程，在这个协商过程中，服务器和客户端会根据各自支持的TLS协议版本来选择一个双方都支持的版本。如果服务器和客户端都支持多个TLS协议版本，那么默认情况下会选择最高的版本。)

2. 证书可靠性

OCSP服务地址如是境外IP，可能会产生相关安全合规问题。OCSP地址如是单IP，极端情况可能会产生运营商封禁IP等导致用户端访问OSCP地址失败从而影响游戏下载。考虑使用OCSP地址是CDN加速的证书可以有效规避以上场景，通过将OCSP地址部署在CDN上，可以利用CDN的冗余和负载均衡机制，提高OCSP服务的可靠性和可用性。即使某个节点出现故障，CDN可以自动切换到其他可用节点，保证服务不中断。

历史上遇到过OCSP服务地址是境外IP，被某地区三大运营商封堵，导致该地区玩家因无法正常完成HTTPS访问影响正常游戏服务。

另外，未开启OCSP Stapling时：客户端的每次请求都会向CA进行OCSP查询，以确认证书未被吊销，频繁的OCSP查询请求导致TLS握手效率较低，将影响用户访问速度。CDN支持OCSP Stapling，开启OCSP Stapling功能后，OCSP信息查询的工作将由CDN服务器完成。CDN通过低频次查询，将查询结果缓存到服务器中（默认缓存时间60分钟）。当客户端向服务器发起TLS握手请求时，CDN服务器将证书的OCSP信息和证书一起发送给客户端，无需再向数字证书认证机构（CA）发送查询请求。极大地提高了TLS握手效率，节省了证书验证时间。

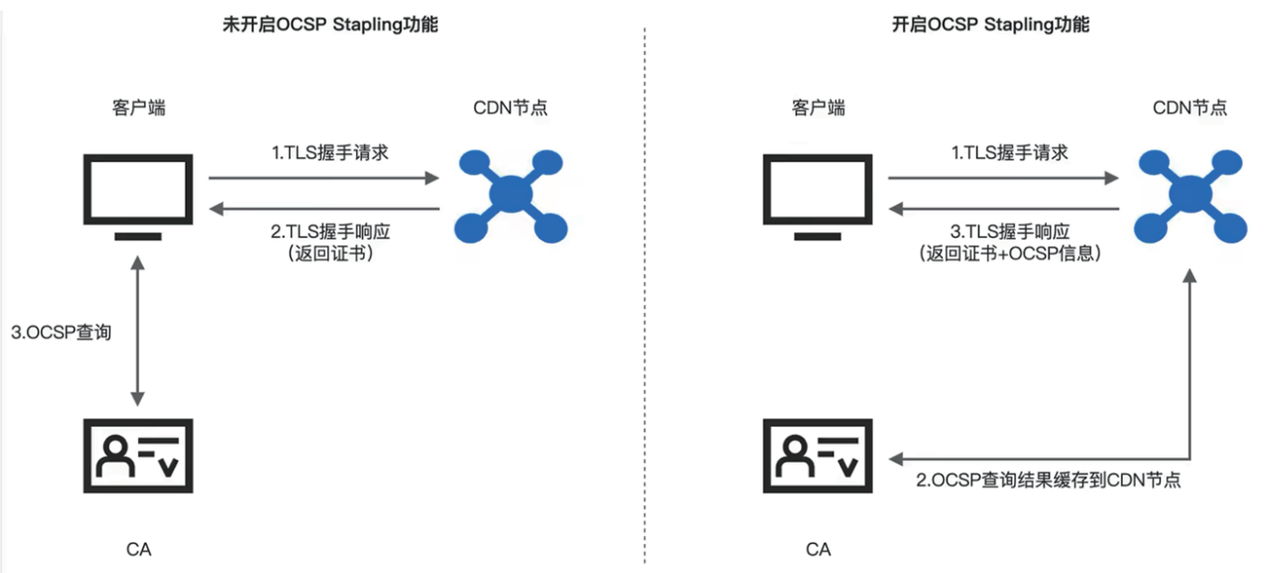

图37

3. HTTPS回源

CDN支持HTTP回源或HTTPS回源，如果使用HTTP回源，有可能存在数据被劫持、篡改风险从而导致CDN缓存异常数据。从安全角度考虑，推荐配置HTTPS回源(需要源站配置HTTPS证书并管理证书更新)。

4.3 配置巡检

4.3.1 基础配置巡检

1. CDN加速区域配置

加速区域选择错误影响加速效果，根据实际业务场景和需求选择正确的加速区域，"全球"、"国内"、"纯海外"。

2. CDN域名OSS类型的源站配置一致

阿里CDN设置OSS为源站时，请在CDN源站类型里选择源站类型为"OSS域名",请勿选择"源站域名"选项。如果选择源站类型为OSS域名，则阿里CDN回OSS时会带上OSS特殊鉴权头，OSS可以此识别是来自阿里CDN的请求，流出流量会按照"CDN回源流量"计费；如果源站是OSS，但是选择源站类型为"源站域名"，CDN回源不会带上OSS特殊鉴权头，OSS无法识别此流量是来自阿里CDN的请求，流出流量会按照"OSS公网流出流量"计费，带来成本差异。

3. 指定源站回源HOST

当CDN加速域名配置了多个源站时，如果继续使用默认回源HOST，使得访问所有源站时都会携带默认回源HOST功能配置的域名，这要求所有源站都要配置对应域名的虚拟站点。指定回源HOST功能可针对不同的回源站点配置不同的回源HOST，能够灵活应对更加复杂的业务场景。在加速域名存在多个源站的情况下，通过指定源站回源HOST功能可以给不同的源站配置不同的回源HOST。

4. 私有Bucket回源

如果CDN加速域名的源站使用的是阿里云对象存储OSS，并且OSS的Bucket被配置为私有模式（可以起到访问鉴权的作用，避免非授权的请求盗刷流量），该情况下建议给加速域名开启OSS私有Bucket回源功能，可以实现通过CDN加速OSS私有Bucket资源。

5. IPv6开关

阿里云CDN大部分节点已支持IPv6协议请求。开启IPv6后，用户在IPv6环境且就近CDN节点支持IPv6时，可通过IPv6协议访问CDN节点；若就近节点不支持IPv6，客户端仍可通过IPv4协议访问。如果有IPv6的接入需求，建议提前开启。

4.3.2 缓存优化配置

1. 设置CDN缓存

CDN缓存命中率低会导致源站压力大，静态资源访问效率低。提升缓存命中率，直接从缓存中获取资源返回给用户，可避免通过较长的链路回源，提高资源的响应速度和降低源站的带宽压力。如果CDN缓存命中率低，会影响用户体验和增加源站的带宽压力。游戏包体由于是不常更新的静态文件，建议设置1个月以上(根据游戏版本更新周期来评估)。

2. 配置忽略参数

当URL请求中带有queryString或其他可变参数时，访问同一个资源的不同URL（URL携带的参数不同）会重新回源，导致CDN缓存命中率低。开启忽略参数功能后，CDN节点在处

理用户请求时，会去除请求URL中携带在?之后的参数（例如：用户身份信息、访问渠道信息），以原始URL来生成缓存hashkey，提升缓存命中率，可根据实际业务场景选择是否开启。

3. 开启Range回源

Range回源可有效提高文件分发效率，可以提高缓存命中率，减少回源流量消耗和源站压力，并且提升资源响应速度。游戏包体都是大文件，推荐开启Range回源，设置为“开启Range回源”（大文件场景推荐配置）。

4.3.3 其他特殊优化配置

如有特殊性能/质量优化需求，请联系对应技术支持服务人员沟通

4.3.4 特殊配置巡检

1. OSS存储空间关闭CDN缓存自动刷新

OSS的CDN缓存自动刷新功能，使用的是OSS大账号下刷新CDN缓存的qutoa，有可能会因为qutoa等原因导致延迟刷新或刷新失败，相对不可控。重点业务建议关闭OSS的CDN缓存自动刷新功能，直接使用CDN的控制台或集成API自动化任务去刷新缓存，相对更可控。

2. OSS生产Bucket和测试Bucket隔离

因Bucket的带宽默认是Region级别的限制，QPS默认是账号级别的限制。如将生产Bucket和测试Bucket放在同账号同地域，则可能因为测试Bucket因测试占用过高的带宽或QPS，从而影响生产环境Bucket的访问，建议分布在不同的UID下，如果您的业务有更高的带宽相关需求，请联系技术支持与服务人员沟通。

4.4 运维监控

4.4.1 客户端埋点日志

1. 特定响应头区分厂商

埋点日志可以记录客户端运行过程中的关键信息，如错误、异常、性能指标等。当客户端发生故障或出现问题时，可以通过分析埋点日志来定位问题的根源，帮助运维人员快速排查和解决问题。对于多CDN服务商的架构，可以在CDN上配置自定义响应头，设置自定义特征值，以快速定位异常来源哪个服务商。

2. 全链路日志追踪

通常需要唯一锁定一条异常请求，则需提供请求时间、客户端IP、服务端IP、请求URL等相关字段，基于这些字段进行日志分析来进一步锁定，沟通成本和排查成本高。而阿里云CDN默认的响应头有一个名称为EagleId的Header，该响应头的Value值是一个唯一标识。这个EagleId可以理解为是一个TraceId或者RequestId，阿里云CDN可以根据这个标识去唯一查到这一次请求对应的全链路CDN日志，包括L1日志、L2日志、回源日志。客户端埋点日志可以考虑捕获该响应头的Value进行上报，方便快速锁定日志进行问题分析，提高排查效率。

3. mtr探测

公网上存在各种复杂的网络问题，在遇到玩家客诉时通常需要具体分析，通过记录客户端ISP信息以及异常访问时客户端进行mtr探测，讲数据埋点上报清洗，可快速分析异常来源是偶发玩家还是面积性的失败，根据mtr数据可快速锁定是否运营商问题，为报障运营商提供关键有效信息。

4.4.2 服务端日志

1. CDN日志

开通CDN日志，可以有效分析CDN日志及时发现问题，并有针对性的解决问题，提升CDN服务质量。阿里云CDN提供了离线日志和实时日志两种功能：

1.1 离线日志：默认支持下载30天内的日志数据(不计费)，如果需要存储更长时间的日志，可以将日志转存到OSS进行长期存储；另外日志延迟较大，通常情况下延迟在24小时之内，也有可能超过24小时。离线日志目前只支持日志转存到OSS云存储，尚未打通日志分析能力。

1.2 实时日志：为实时采集的日志数据，日志数据延迟不超过3分钟，而离线日志的数据延迟通常在24小时之内。CDN实时日志打通了SLS日志服务的日志转存和日志分析能力，预制了CDN基础数据、CDN错误分析、CDN热门资源、CDN用户分析这四张常用分析报表，也支持自定义日志分析策略，可以一站式提供日志存储和日志分析能力。实时日志会产生实时日志投递费用和SLS的费用。

2. OSS日志

访问对象存储OSS的过程中会产生大量的访问日志。实时日志查询功能将OSS与日志服务SLS相结合，允许用户在OSS控制台直接查询OSS的访问日志，帮助用户完成OSS访问的操作审计、访问统计、异常事件回溯和问题定位等工作，提升工作效率并更好地帮助用户基于数据进行决策。如果需要更细致地追踪和分析用户行为或满足特定的监控要求，可能需要设置其他字段。OSS的自定义日志字段user_defined_log_fields允许记录特定的请求头和查询参数，以满足自定义字段分析需求。

4.4.3 监控告警

1. CDN监控告警

CDN告警指标，可以依赖云监控指标进行告警配置。包括活动域名带宽告警，整体带宽水位超过80%、85%、90%、95%后触发告警，明确做预案准备，比如：客户端调整占比。

2. 活动域名4xx\5xx监控

表17

MetricName	指标类型	MetricDescribe	Dimensions	Statistics	Unit	MinPeriods	是否可以设置报警
BPS	实例维度	带宽峰值	userId,instanceId	Average,Minimum,Maximum	bits/s	60 s	是
CodeRatio404	实例维度	边缘状态码404占比	userId,instanceId	Average,Minimum,Maximum	0	60 s	是
CodeRatio416	实例维度	边缘状态码416占比	userId,instanceId	Average,Minimum,Maximum	0	60 s	是
QPS	实例维度	每秒访问次数	userId,instanceId	Average,Minimum,Maximum	count	60 s	是
code4xx	实例维度	边缘状态码4XX占比	userId,instanceId	Average,Minimum,Maximum	0	60 s	是
code5xx	实例维度	边缘状态码5XX占比	userId,instanceId	Average,Minimum,Maximum	0	60 s	是
code_ratio_499	实例维度	边缘状态码499占比	userId,instanceId	Average,Minimum,Maximum	0	60 s	是
hitRate	实例维度	边缘字节命中率	userId,instanceId	Average,Maximum	0	60 s	是
ori_bps	实例维度	回源网络带宽	userId,instanceId	Average,Minimum,Maximum	bits/s	60 s	是
ori_code_ratio_4xx	实例维度	回源状态码4XX占比	userId,instanceId	Average,Minimum,Maximum	0	60 s	是
ori_code_ratio_5xx	实例维度	回源状态码5XX占比	userId,instanceId	Average,Minimum,Maximum	0	60 s	是
rt	实例维度	边缘响应时间	userId,instanceId	Average,Minimum,Maximum	ms	60 s	是

3. OSS监控配置

表18

MetricName	指标类型	MetricDescribe	Dimensions	Statistics	Unit	MinPeriods	是否可以设置报警
Availability	cms.metric.metricGroup.Bucket	可用性	userId,BucketName	Value	0	60 s	是
CdnRecv	cms.metric.metricGroup.Bucket	cdn流入流量	userId,BucketName	Value	B	60 s	是
CdnSend	cms.metric.metricGroup.Bucket	cdn流出流量	userId,BucketName	Value	B	60 s	是
GetObjectE2eLatency	cms.metric.metricGroup.Bucket	GetObject请求平均E2E延时	userId,BucketName	Value	ms	60 s	是
GetObjectServerLatency	cms.metric.metricGroup.Bucket	GetObject请求平均服务器延时	userId,BucketName	Value	ms	60 s	是
InternetRecvBandwidth	cms.metric.metricGroup.Bucket	公网流入带宽	userId,BucketName	Value	bit/s	60 s	是
InternetSendBandwidth	cms.metric.metricGroup.Bucket	公网流出带宽	userId,BucketName	Value	bit/s	60 s	是
IntranetRecvBandwidth	cms.metric.metricGroup.Bucket	内网流入带宽	userId,BucketName	Value	bit/s	60 s	是
IntranetSendBandwidth	cms.metric.metricGroup.Bucket	服务端错误请求占比	userId,BucketName	Value	bits/s	60 s	是
ServerErrorRate	cms.metric.metricGroup.Bucket	成功请求占比	userId,BucketName	Value	0	60 s	是

MetricName	指标类型	MetricDescribe	Dimensions	Statistics	Unit	MinPeriods	是否可以设置报警
TotalRequestCount	cms.metric.metricGroup.Bucket	总请求数	userId,BucketName	Value	count	60 s	是
UserAvailability	cms.metric.metricGroup.userId	用户层级可用性	userId	Value	0	60 s	是

4. OSS事件订阅

OSS提供了用户级别和Bucket级别的流控，支持的类别主要包括带宽流控和QPS流控。当访问OSS的QPS、带宽超出OSS使用限制时，访问速度会受到OSS流控的限制。如果触发了带宽流控，则访问OSS的延迟会增加。如果触发了QPS流控，则OSS会丢弃部分请求。可以通过云监控管理控制台创建OSS流控事件告警规则，并指定在监测到用户发送到OSS指定类型的请求量触发流控或达到汇报阈值时，以短信、邮件和钉钉机器人的方式向指定联系人组发送报警信息，事件订阅支持降噪功能。具体请参见通过云监控服务实时监控OSS流控信息。

4.5 容灾预案

4.5.1 限流预案

1. CDN量级调整

对业务域名进行水位监控，整体带宽水位超过80%、85%、90%、95%后触发告警，明确做预案准备。如果某一厂商带宽水位超过100%且无法承接更多带宽，将多余的量切到其他厂商。如果有友商故障需要切到阿里云时，请务必提前同步阿里云评估（量级、流数），由阿里评估承接。

2. CDN带宽限流

用户临时突增流量超过CDN厂商服务能力上限，在无法承接更多量级的情况下，为保障整体服务稳定性，需要CDN厂商以域名为粒度进行限流，对域名的全网L1出带宽施加控制。针对域名设定一个限流阈值，将域名L1出带宽控制在阈值左右，正负5%上下波动。限流功能是调节反馈机制，因此对于流量突增场景需要一定的调节周期才会将带宽控制在阈值上下，对于突增流量，带宽超过阈值5%属正常情况。这里给出的正负5%浮动，只针对QPS平稳场景，不适用于QPS抖动场景。

- 当域名带宽低于限流阈值时，对请求不做任何处理。
- 当域名带宽超过限流阈值时，对所有请求进行限速处理，限制速率会随着带宽的上下波动做周期性的动态调整，随着动态控制请求的发送速率，域名总带宽会被控制在阈值左右。

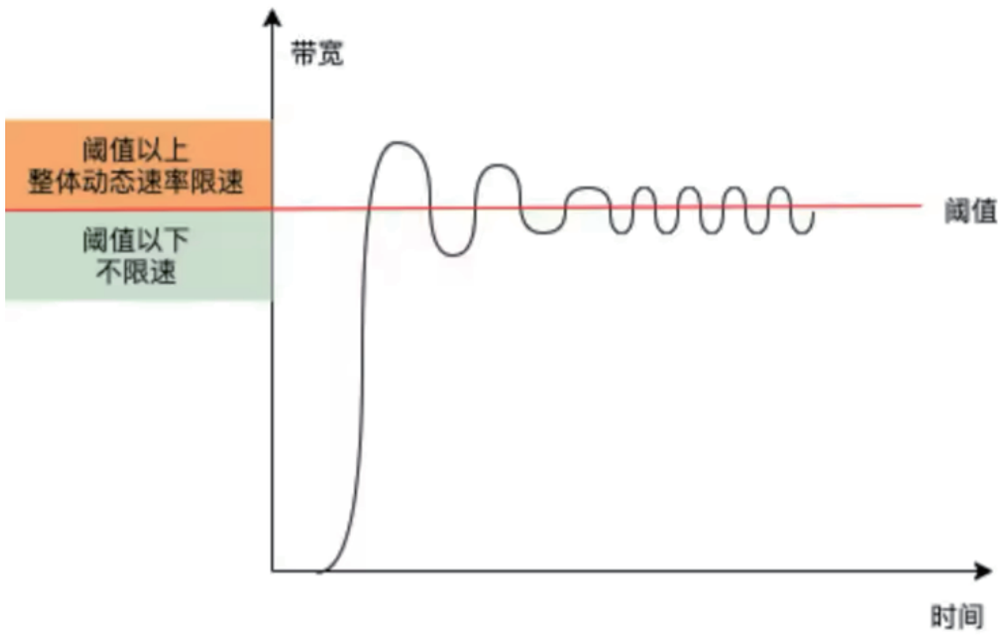

图38

4.5.2 质量预案

区域性质量下降

客户区域性业务访问质量下降，具体表现为499/5xx等状态码异常升高，剔除异常节点，排查节点网络整体稳定性无异常，域名指标异常持续发生，更换调度节点(调度调整后，因DNS缓存问题DNS调度预计需要10min生效，持续观察)。

4.6 活动准备

4.6.1 CDN资源报备

游戏资源包含各渠道的游戏首包、更新包，serverlist文件、其他静态文件等。由于现在游戏包体越来越大，因此在游戏OB前一般会先开放预下载，同时也会配合运营活动宣传。如玩家预约量上涨，则开放下载后可能带来更大的突发下载带宽。因此对于预下载和正式开放下载的CDN带宽评估非常重要，客户将评估的CDN峰值带宽和QPS提前报备给CDN服务商，活动时间需要精确到小时及预估峰值时间，CDN服务商依赖该信息做资源预留，避免CDN资源不足导致的下载速度慢、甚至下载失败的情况。

通常带宽的预估由客户完成，因为涉及客户的一些业务数据等，但有时候可能客户也需要我们的协助，或者我们也要协助判断下客户预估数据的合理性(是否测算错误，带来较大水分等)。通常这需要收集一些基础数据，包括预约玩家数量、游戏包体大小、历史数据参考、用户行为等。峰值带宽=同时下载的用户数×每个用户的平均下载速度。通常在开放下载后的前几个小时内，会有一个非常高的下载峰值，接下来的几小时或几天内，下载量会逐渐减少但仍然较高，之后下载量会趋于平稳。如果是把大的包体切分成多个小文件(如Steam等)，则QPS会显著增加，这种场景下QPS需要重点评估和报备QPS。

4.6.2 CDN质量优化

对涉及到的CDN相关产品进行配置巡检和质量调优，具体参考前文的配置巡检篇。

4.6.3 OSS资源报备

1. OSS限制说明

● OSS上下行带宽限制：默认情况单个阿里云账号在中国内地和非中国内地的各地域的上下行带宽(区分内外网)均有限制，具体请参考官网"OSS使用限制及性能指标"产品限制说明。如果达到阈值，请求会被流控。当请求被流控时，请求返回的Header中会携带x-oss-qos-delay-time: number。其中number为请求被流控的时长，单位为ms。上传类请求会返回精确的被流控的时长；下载类请求会返回根据流控程度和文件大小估算出的被流控的时长。带宽流控影响的是速度，但不断开TCP连接、不对请求置错，只是带宽不能继续打上去了。

● OSS QPS限制：单个阿里云账号的总QPS为10,000，但在不同的读写方式下，实际能达到的值也不同。具体也参考"OSS使用限制及性能指标"。QPS流控会报错TotalQpsLimitExceeded。OSS根据文件key的UTF-8编码顺序自动划分数据分区，以支持大规模文件管理和高并发请求。然而，在使用顺序前缀（如时间戳或按字典序排列的字符串）的情况下，可能引起部分分区过载现象，即大量文件集中在少数几个分区中。为了优化OSS的数据分布和提升处理效率，建议采用随机性前缀替代传统的顺序前缀来命名文件，具体参考OSS性能最佳实践。

2. 具体场景

● 上传场景(内网上行)：游戏服务日志定时上传OSS，会有内网上传带宽和qps的突发，特征表现为定时突刺型(毛刺)，持续时间短。通常游戏服务器上，上传日志服务进程也有一定限制，不会对服务器的性能有太大占用，通常可以接受瞬时/短时间的带宽流控，但如果业务并发设计不合理、进程设计不合理、带宽阈值设置较低，如果长时间流控，可能会对服务产生一定风险。

● 下载场景(公网下行)：通常在开放下载前，会提前几天上传包体到OSS，然后在CDN侧进行预热。CDN预热时会有OSS带宽的突增，需要控制好预热的并发，避免把OSS带宽打的太高。正常情况在开放下载以后，因为热度高，都命中CDN的缓存，OSS的带宽和QPS预期内压力不大。在热更场景里，需要临时发布一个版更进行热更(紧急修复等场景)，会有紧急的回源，此时也会有OSS带宽的突增。

● 大数据场景(内网下行)：大数据业务定时从内网批量下载数据用于大数据作业分析，会有突发的内网下行带宽，可能会持续一段时间。因为是离线大数据作业，通常即使内网下行带宽流控，也不影响业务，具体以客户业务场景为准。这种情况需要考虑云监控流控事件告警的降噪。

3. OSS带宽

● 基于客户不同业务场景下，不同的Bucket对内网上行、内网下行、公网上行、公网下行的不同需求，进行带宽、QPS的需求评估，评估是否针对具体Bucket进行Qos调整。带宽通

常与存储量有比例换算，如果超出比例的部分则需要评估付费，不同地域的带宽能力也有所差异，需要统一和技术服务做好沟通。

4.6.4资源预热

- 通过预热的方式，提前将资源分发到全球各个CDN节点进行缓存。这样，在活动开始前CDN节点就已经缓存了大部分或全部的资源，用户访问时可以直接从就近的节点获取，减少网络延迟和提高访问速度。预热时需要考虑节点负载和源站负载，根据包体大小、源站负载等实际情况合理控制预热并发，避免预热并发较大导致源站限流等异常，因此通常需要提前几天报备资源进行预热。客户自助控制台或者API预热可以预热到L2节点，如果需要预热到L1节点，需要厂商后台预热，可以根据包体大小、活动重要程度、对下载速度要求综合考虑决策。

多CDN服务商的架构下，需要确认CDN友商也控制预热并发，避免多个厂商同时预热产生源站带宽压力较大的情况。

第五章 游戏数据库：瞬息恢复的艺术

在游戏行业，玩家数据属于核心资产，在日常维护和版更时数据的变动会非常大，除了要求运维开发严格按照规范流程变更外，还需要随时具备快速恢复数据的能力。经常会遇到用户误操作或者程序缺陷导致的数据问题，需要通过已有的技术手段帮助客户完成数据保全或者尽量减少数据损失。希望在这种场景下能更好地应对数据应急回档场景的问题，确保系统的稳定运行和客户的良好体验。

5.1 灾难的发生

关于数据丢失灾难的发生场景非常的多，为了帮助大家有效识别将云上目前遇到的几种较常见的场景进行罗列提供大家进行参考，这里后续还会继续基于发生的情况继续迭代。

5.1.1 资源生命周期问题

线上核心资源是否已经设置自动续费，在云上每年都会因为各种原因出现几次账号欠费，未自动续期等导致实例释放的问题。如果是线上业务出现欠费情况对业务将会是一个严重灾难。

5.1.2 实例误操作释放

几乎每年都会看到有客户因为运维或者开发操作失误导致了线上核心资源的释放而导致重大业务故障。有出现批量释放错误勾选，有手抖误选，有业务标签别名标记不清等导致的误操作释放不胜枚举。

5.1.3 错误配置变更（程序错误）

错误配置导致数据错误的场景遇到的情况也非常多，比如有遇到过update 更新的时候没有忘记加上谓词全表生效这种场景。

5.1.4 误操作删除数据

业务开发或者运维错误操作delete、drop或truncate等导致的数据丢失，这种场景一般出现在版更停服维护期间，由于操作人员未精准判断delete范围或者误操作了数据的清理或者drop导致数据丢失。

5.2 恢复原理介绍

基于上文出现的常见造成数据丢失的场景，通常的数据恢复方案主要包含资源实例层级恢复和SQL层级恢复两大类。

5.2.1 实例层级恢复

一般对于资源忘记续期到期释放或者误操作提前释放场景，云数据库提供了回收站功能。通常来讲线上核心的数据库我们都会建议进行自动续期都选，但是意外情况发生一般业务都会较快感知到业务异常，通过回收站找回实例，以RDS为例，实例因为到期被锁定，7天内该用户可以在回收站中对实例进行续费解锁。超过7天后只能在回收站中对实例进行重建恢复，将原实例的数据恢复到一个新的实例上。

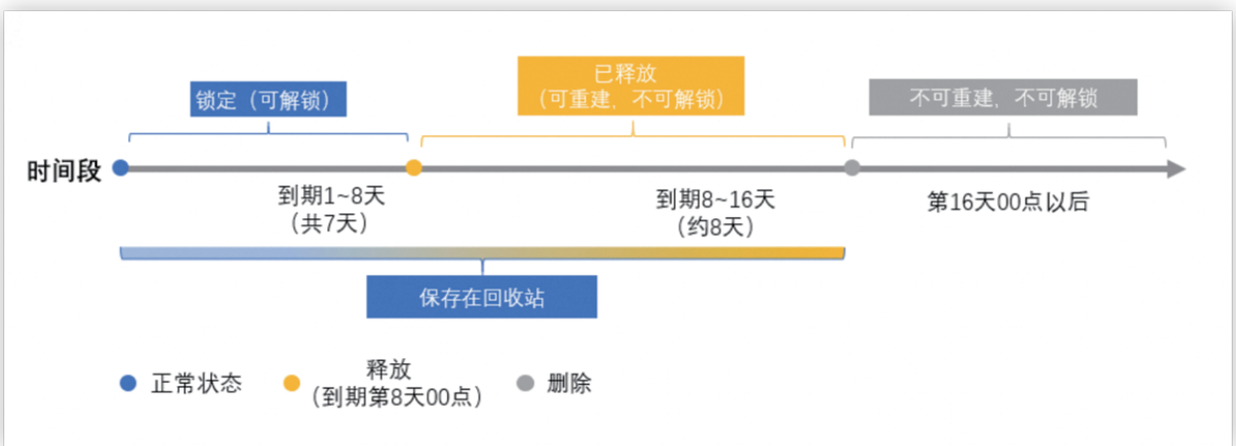

图39

如果是到期锁定7天内解锁到原实例恢复，超过7天后将需要进行重建，重建过程中用户需要重点关注创建时的VPC信息，确保创建出来的资源能正常和业务交互。

5.2.2 SQL层级恢复

针对于运维开发人员SQL不规范或者程序逻辑异常导致的数据丢失，通常云资源实例还存在，这个时候我们需要基于实例全量和增量备份来恢复数据。

举一个案例：比如游戏停服时间段在早上6点-10点，运维在停服后大概在在7点左右按照业务既定计划进行数据历史数据清理，清理完毕后QA同学在9点左右测试发现大量玩家数据不存在，判断数据出现严重丢失。根据运维操作记录查看发现运维操作delete过程中没有填写判断条件执行了全表删除，实例上所有玩家数据丢失。

这个时候客户侧就需要快速找回删除前的数据。对于关系型数据库支持按照时间点恢复，对应实现原理为基于离恢复时间点最近的全量备份+增量Binlog（或者Redolog），按照指定时间点增量回放。针对这种场景操作需要非常谨慎，因为操作错误会大大影响业务恢复时长，一般我们需要重点关注一下几点：

1. 定位异常发生准确时间点log

基于备份恢复的方案需要准确获取业务操作的时间点，如果选择时间早了存在部分写入数据丢失，如果恢复时间选择过于靠后，恢复出来的数据已经发生故障则恢复无意义，所以对于定位故障发生时间点非常重要。定位工具一般包括慢日志，审计日志和Binlog日志。慢日志和审计日志比较轻量级，如果慢日志有记录会比较快速定位时间点，但是需要重点留意慢日志记录时间为SQL完成时间，有些场景SQL执行较快未记录慢日志可以尝试通过审计日志进行定位发生时间点，速度相对也较快。有些情况从成本考虑没有开启洞察审计功能，这里需要通过下载Binlog解析完成时间点定位，操作成本和量级相对比较高，建议业务程序可以提前进行演练保留对应常见的检索方案以及脚本方便快速定位。如果业务有对应工具记录操作时间点同样也可以实现快速定位，比如通过DMS操作可以记录变更时间等类似方案。

2 选择恢复方案

恢复方案包含3种，一般分为恢复到原实例，恢复到新实例，恢复到本地。通常来讲数据出现异常场景业务恢复比较着急，选择恢复到原实例速度相对较快所以这种方式选择较多，恢复出来假设原表为table1,新表将默认为table1_bak，按照时间点恢复业务验证数据符合预期后进行rename table交换表名操作即可提供业务使用。

如果选择恢复到新实例或者本地的话，这种一般适合数据变更较少场景，通过新的数据库记录在原实例订正，可以通过DTS等方案来进行数据回迁，也可以手动订正。

5.3 演练与实操

以Polardb为例进行实际演练和操作

5.3.1 演练步骤：

- DBA 在生产实例上创建用于演练需要的数据库和表，并装载测试数据
- DBA 人为删除演练用的测试数据模拟数据灾难
- 确定误删除时点，并确定对当前应用无影响，开始操作恢复
- 确定需要恢复到的时间点，时间点由开发给出或者 DBA 在 Binlog 中查找误操作的位点
- 在阿里云控制台按时间点恢复的流程执行操作创建恢复实例
- 对恢复实例中的数据进行验证确定丢失数据已找回
- 将恢复实例的数据导回到生产实例

5.3.2 人为删除数据模拟灾难

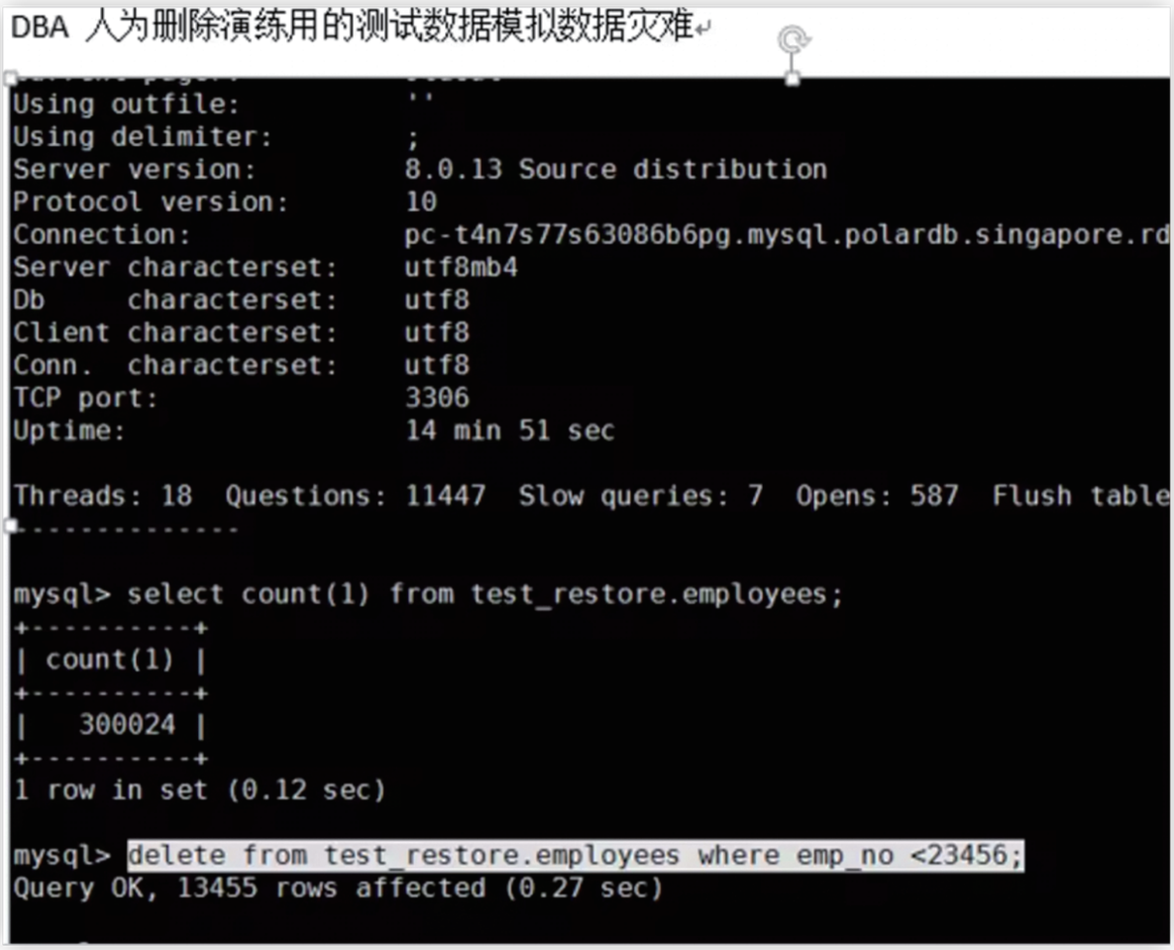

图40

用户查找删除的命令：delete from test_restore.employees where emp_no<23456

图41

解析Binlog查看发生时间情况

```
#200311  6:08:26 server id 100302488  end_log_pos 15679952 CRC32 0x5a73ad77     Delete_rows_v1: table id 88
# at 15679952
#200311  6:08:26 server id 100302488  end_log_pos 15688140 CRC32 0xe6c532d1     Delete_rows_v1: table id 88
# at 15689140
#200311  6:08:26 server id 100302488  end_log_pos 15689158 CRC32 0x7e5cf883     Delete_rows_v1: table id 88 flags: STMT_END_F
### DELETE FROM `test_restore`.`employees`
### WHERE
###   @1=10001 /* INT meta=0 nullable=0 is_null=0 */
###   @2='1953:09:02' /* DATE meta=0 nullable=0 is_null=0 */
###   @3='Georgi' /* VARSTRING(42) meta=42 nullable=0 is_null=0 */
###   @4='Facello' /* VARSTRING(48) meta=48 nullable=0 is_null=0 */
###   @5=1 /* ENUM(1 byte) meta=63233 nullable=0 is_null=0 */
###   @6='1986:06:26' /* DATE meta=0 nullable=0 is_null=0 */
### DELETE FROM `test_restore`.`employees`
### WHERE
###   @1=10002 /* INT meta=0 nullable=0 is_null=0 */
###   @2='1964:06:02' /* DATE meta=0 nullable=0 is_null=0 */
###   @3='Bezalel' /* VARSTRING(42) meta=42 nullable=0 is_null=0 */
###   @4='Simmel' /* VARSTRING(48) meta=48 nullable=0 is_null=0 */
###   @5=2 /* ENUM(1 byte) meta=63233 nullable=0 is_null=0 */
###   @6='1985:11:21' /* DATE meta=0 nullable=0 is_null=0 */
### DELETE FROM `test_restore`.`employees`
```

图42

5.3.3 核对Redolog轮转原理，选择合适时间点

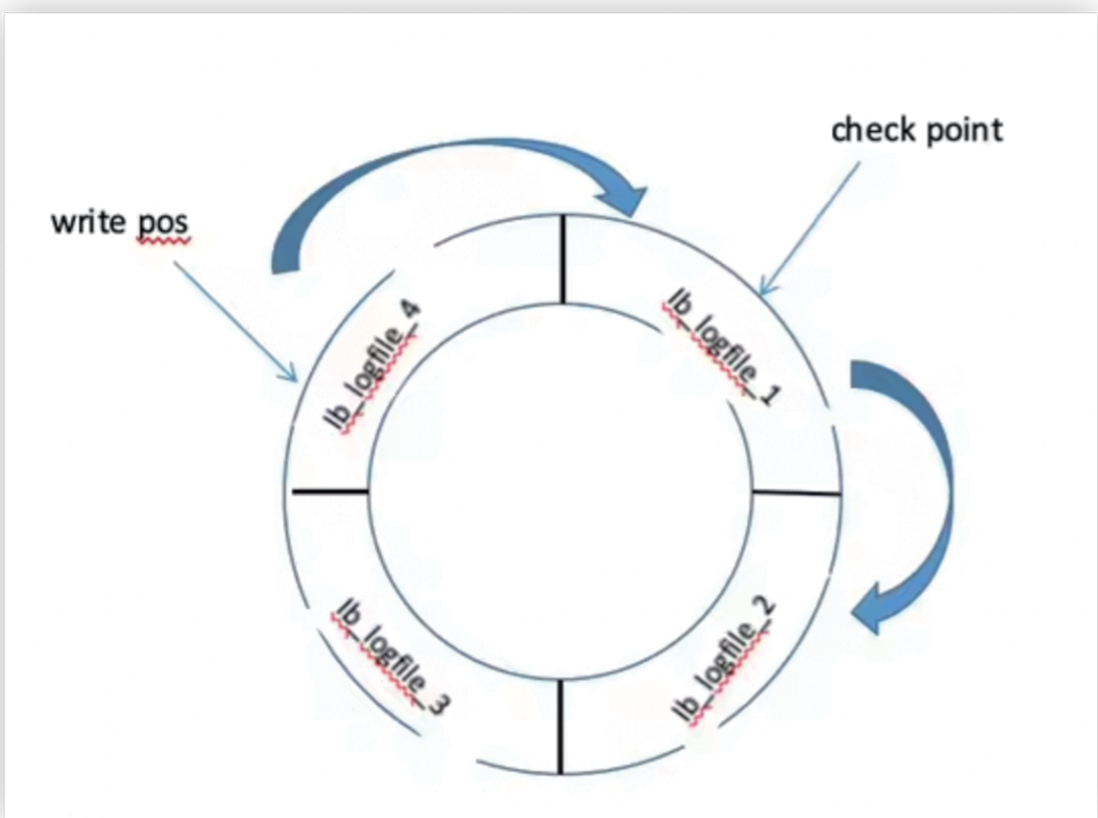

图43

5.3.4 进行数据恢复并订正数据

1.进入Polardb控制台找到对应实例ID，点开实例详情，选择备份恢复栏，选择按时间点恢复

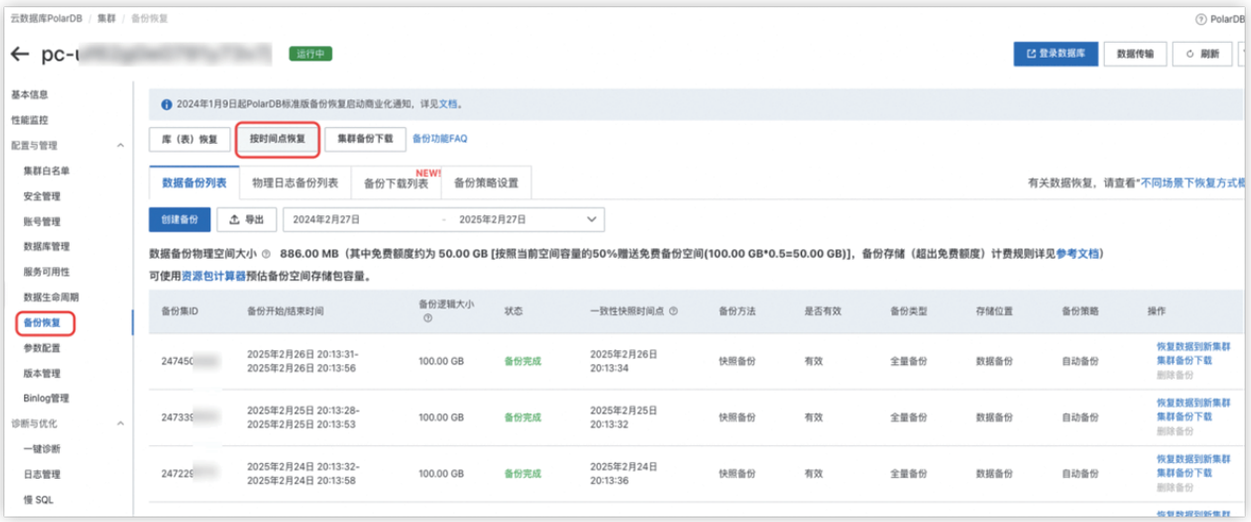

图44

2.选择恢复的时间点和目标的VPC网络（关键步骤）

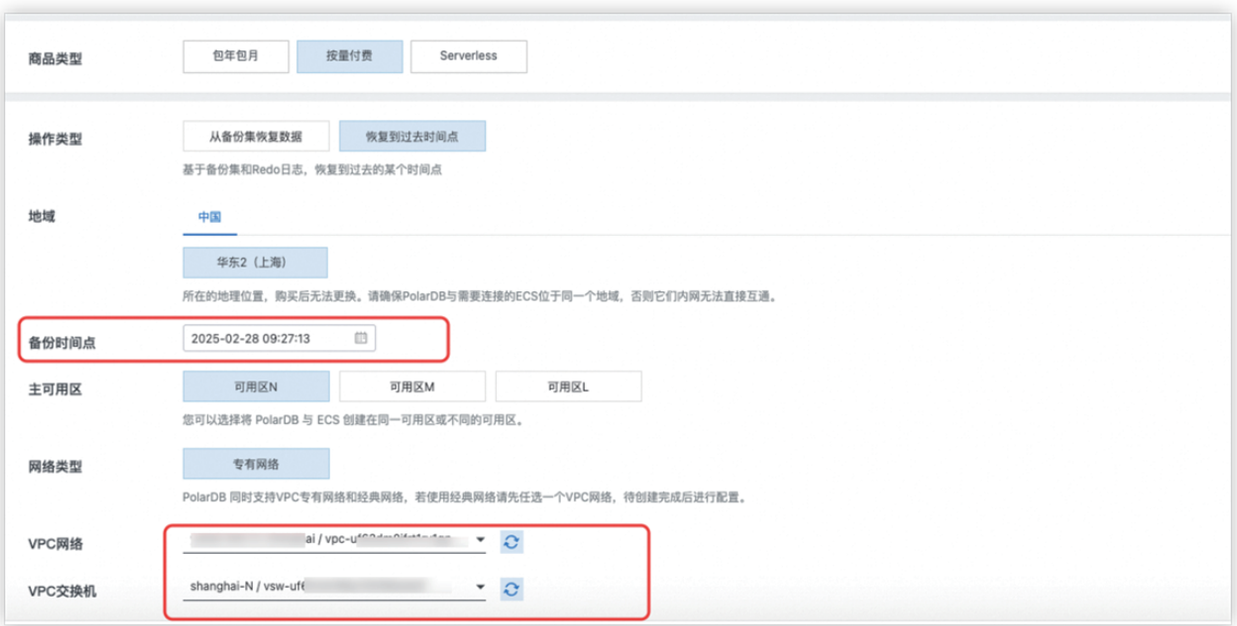

图45

3.待实例恢复正常后登录数据库验证数据，如果符合预期通过手动订正，或者DTS迁移方案迁移到源库，至此数据完成恢复。

5.4 规避数据损失方案

事后补救始终是有损方案，所以这里帮助大家梳理下对于数据保护的一些常见的措施，可以给大家一些参考建议，希望对大家有一定的帮助。

5.4.1 账号维度

账号上配置账号费用阈值报警配置和延停配置，避免因为欠费导致实例异常。

5.4.2 实例维度

核心业务实例配置自动续费和释放保护，避免因为未及时续费导致实例被终止服务或者释放。

5.4.3 权限管控

准确做好权限管控，按照最小权限原则设置数据库和RAM权限，避免部分人员误操作导致实例或者数据异常。

5.4.4 定期备份

业务重大变更前的备份，方便加速数据恢复速度。

5.4.5 依托云产品高频备份加速恢复

业务评估开启高频备份功能，降低全量+增量追数据耗时，较少业务损失时长。

5.4.6 依托云产品审计审批功能

建议业务开启DMS数据库管理，针对数据库的操作已经有了完整了审计和安全审批流程。大部分企业的数据安全都是形同虚设，数据安全也是典型的重要不紧急工作，但出问题了就是致命一击，所以必要的安全审批能很大程度减少这种损失。

5.4.7 跨地域灾备

跨地域灾备可以在极端不可抗力因素下确保数据能够正常提供服务，给数据做双重保障，确保业务能够快速恢复。

5.5 结语

随着游戏业务的发展，数据库在游戏场景中的重要性逐步被体现，其中玩家数据作为游戏的生死线尤其重要，我们作为云上游戏客户的守护者，需要除了需要在关键节点能够帮助客户恢复数据完成止血尽量降低业务损失外，还需要积极推动客户进行数据保护和数据丢失预防工作，降低数据风险问题，同时也需要用户进行必要的容灾演练和模拟数据丢失紧急回档演练，建立最后一道防线。

第六章 游戏大数据：探索玩家心声的数据海洋

2024年，中国游戏产品在全球范围内取得较佳表现，国内市场、自研产品出海市场均取得”2024年收入创历史新高”的成绩（国内销售3257.83亿元，增长率7.53%；海外市场销售收入183.57亿元，增长率13.39%）。从细分数据及用户反馈来看，市场规模呈现出由多个细分市场共同促进增长的特征。（伽马数据：2025年中国游戏产业趋势及潜力分析报告）玩家的选择越来越多，需求也在不断变化，如何让玩家和用户（玩家可能是指游戏玩家，但是用户可能包含开发者和社区的用户，TapTap的业务模式就同时包含了这三种角色）愿意留下来，如何提升他们的付费意愿，如何持续推出有吸引力的内容，同时能够实现数据驱动的战略，已经成为每一家游戏公司都必须面对的难题。

过去，游戏公司似乎主要依靠经验和直觉来制定运营策略，比如通过大规模买量获取新玩家，或者依靠人工分析玩家数据来优化游戏体验。然而，随着市场环境的变化，这些传统方法越来越难以奏效。广告成本越来越高，新用户获取越来越难，而玩家的耐心也越来越有限，再者，如果游戏内容、玩法或者活动设计跟不上，即便是老用户也会很快就会流失。所以对于游戏公司来说一方面是运营上要更加的精细化，另一方面也要关注为“精细化”提供底层技术支撑的云产品的能力以及稳定性，这两者缺一不可，相辅相成，如果因为技术底座不稳定或者运营不够精细化以至于不能很好的反馈客观事实，不仅会影响玩家体验，还可能对游戏品牌造成负面影响。

在这样的背景下，利用数据驱动运营，成为游戏企业提升竞争力的关键。阿里云在游戏行业深耕多年，云上大数据产品体系，为游戏公司提供了一整套从数据采集、存储、计算到分析的全链路解决方案，帮助企业构建稳定，可靠，高效的技术底座。举例来说，游戏公司可以通过实时数据采集（实时计算Flink版），精准获取玩家的行为轨迹，从而更快发现玩家偏好，优化游戏内活动和玩法；依托云上湖仓数据处理能力（Maxcompute，Starrocks，Hologress），可以实时监测游戏异常情况，例如突然的玩家流失、服务器负载过高等，帮助运营团队快速响应；智能数据分析（QuickBI）则可以深度挖掘玩家数据，精准识别高价值用户，并通过个性化推荐提升玩家付费转化率，为业务人员提供更为高效，稳定的查询用数体验。

本章我们将从两家游戏公司典型的游戏运营出发（A公司的社区运营&B公司的广告投放平台），介绍底层大数据产品的能力，以及阿里云在游戏行业提供的基于大数据产品解决方案。

6.1 游戏运营场景

6.1.1 游戏网络的社区运营

1. 业务背景

A公司是一家中国领先的游戏开发与发行公司，同时也运营着知名的游戏社区平台，主要业务包括游戏研发与发行和游戏社区平台运营两大板块，而玩家反馈和社区管理是游戏社区平台运营的关键一环，“发现好游戏”，“与玩家站在一起”，这是他们遵循的信念准则，A公司在这方面采取了多项运营策略。

- 玩家反馈处理机制
  - 实时评分监控与干预
    - 每小时监控游戏评分波动，评分低于4.8分时自动触发「福利补发」机制，通过赠送道具或福利稳定玩家情绪
    - 针对差评类型（如崩溃闪退、充值不到账等），预先设计3套标准化的客服话术模板，提升响应效率
  - 分层反馈响应体系
    - 要求开发者和运营团队在24小时内回复玩家评价，通过标准化话术表达重视态度（如“感谢反馈，我们会尽快解决”）
    - 对负面反馈采取“真诚倾听+解决方案”策略：优先承认问题并说明改进计划，避免与玩家发生冲突
- 社区运营核心策略
  - 用户分层管理

将玩家分为核心玩家、KOL、泛用户三类，为核心玩家提供专属福利（如测试资格），鼓励KOL产出攻略内容，泛用户通过活动提升活跃度

  - 舆情与数据分析
    - 监控社区互动率、日活/月活等核心指标，结合玩家行为数据优化运营策略
    - 通过爬虫抓取实时搜索关键词（如“抽卡模拟器”），快速响应玩家需求并开发配套工具
  - 内容生态建设
    - 策划游戏文化传播活动（如版本更新直播、玩家共创关卡设计），增强社区归属感
    - 建立玩家投稿激励机制，优质攻略可导流至付费专栏实现创作者变现

2. 业务痛点

A公司的业务需求痛点主要集中在三个方面：

- 一是如何更好地赋能核心业务，包括用户运营、发现好游戏和广告营收；
- 二是如何实现数据统一，让社区，游戏，以及开发者有一个统一的OneID，减少数据烟囱和孤岛现象，通过统一元数据管理和数据服务提升数据治理能力；
- 三是如何在激烈的竞争中降本增效，包括降低运维和资源成本、加快数据开发效率以及提供更高效的自助化服务。

在业务特征方面，A公司专注于高质量游戏推荐，通过编辑精选和算法推荐为玩家提供个性化游戏体验，尤其重视独立游戏和小众精品；同时，支持独立开发者实现低门槛入驻和公平竞争，并借助数据分析和反馈工具帮助开发者优化游戏；此外，A公司将游戏分发与社区功能相结合，为玩家提供一站式体验，支持从游戏发现、下载到评价、讨论的全流程，并通过社交互动增强玩家之间的联系；最后，A公司以数据驱动运营，深入分析玩家行为，优化推荐算法和运营策略，同时为开发者提供市场趋势洞察，助力游戏生态的持续发展。

3.A公司数据中台-打通游戏和开发者平台以及社区

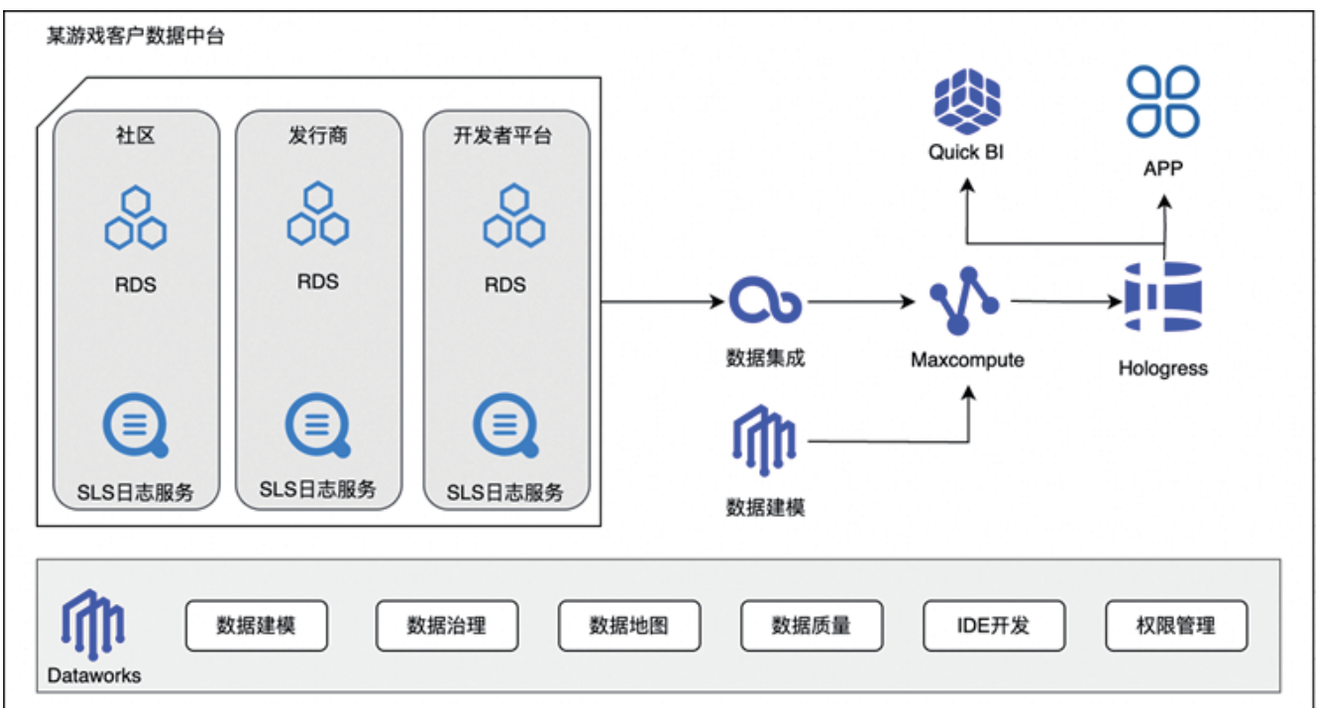

图46

6.1.2 游戏广告投放

1. 业务背景

B公司成立于2017年，是一家独特且创新的游戏公司，目前在全球已有超过1亿的手机游戏用户。数据分析场景集中在广告投放平台用户分析，游戏推广期间每分钟会产生上万美元广告费用，实时反馈可以尽可能降低无效投放费用。

B公司对于广告投放平台的需求，一句话总结就是——“如何提升广告投放效果”，这方面对精细化运营提出了更高的要求，一方面要搭建一套完善实时竞价系统（RTA平台）判断出合适出价，提升流量采买效率。另一个重要方面就是通过游戏广告分析平台同时细化目标人群标签维度，构建社区内容加强私域运营，提高用户识别能力，将广告投放的营销相关的数据沉淀在自己的广告分析平台中，将营销数据与游戏业务内的数据相结合得到全面的用户闭环数据，判断买量是否达到了预期目标且在后期游戏内是否有效转化，保障游戏运营进行更深入的分析迭代出最佳投放方案，再进一步进行精准投放降本增效以及不断优化提升广告转化效果，而建设数字化的广告投放平台是精细化投放的必由之路。

2. 业务痛点

从业务需求的角度来看，B公司作为一家专注于高质量游戏开发和发行的游戏公司，其广告买量的核心需求是通过精准的广告投放和高效的用户触达，最大化游戏的下载量和用户留存率。具体来说，B公司需要通过广告买量实现以下目标：一是精准定位目标用户群体，提升广告点击率和转化率；二是优化广告投放策略，降低获客成本，提升广告 ROI；三是通过数据分析和用户画像，实现个性化广告投放，提升用户体验和用户粘性；四是整合游戏、社区和开发者数据，形成数据闭环，为广告投放提供更全面的决策支持。游戏广告平台，从技术的视角分析的痛点有以下几个方面：

- 数据规模超大、数据快速能导入导出：广告投放分析需要对接接入多方的数据库源，并且随着数据源持续导入以及人群分析维度的逐渐丰富，数据规模会持续的放大，广告分析平台必须要能够支撑超大的数据规模；广告投放的数据的来源也是多种多样（日志、文本等）需要快速导入的能力、以及投放圈选结果也需要快速导出给下游投放系统；
- 实时分析计算性能要求高：广告投放分析通常需要在亿级别的人群数据中，先进行多种维度的圈选，形成多个打标结果的人群包，然后这些人群包进行任意维度交叉分析秒级返回结果，对计算性能的要求非常高；
- 运营活动之间互相干扰：大的运营活动计算资源需求量大，运行时间超长则会干扰其他正常的运营活动的计算任务，需要广告分析平台能够支持资源的调度隔离等混合负载能力；

从业务特征的角度来看，B公司的广告买量需要结合其高质量游戏推荐和社区运营的特点，打造差异化的广告投放策略。例如，通过 A公司的编辑推荐和算法推荐能力，B公司可以精准筛选出高潜力用户群体，并结合社区互动和玩家行为数据，优化广告内容和投放渠道。此外，B公司还可以利用 A公司的开发者工具和数据分析能力，深入了解用户需求和市场趋势，进一步提升广告投放的精准度和效果。通过将游戏分发与社区功能相结合，B公司可以为广告投放提供更多场景化触点，增强用户与广告的互动性，从而实现更高效的广告转化和用户增长。

3．B公司广告投放平台

B公司的广告投放需求主要集中在三个方面：一是广告投放效果评估，通过多维度数据分析广告的表现和用户反馈，确保广告投放的精准性和效果最大化；二是广告预算分配，基于目标用户群体和投放渠道的性价比，合理分配预算以优化广告投放的资源利用率；三是投放策略优化，通过实时数据监控和用户行为分析，动态调整广告投放策略，提升广告点击率、转化率和用户留存率，从而实现高效的广告投放效果和资源回报。

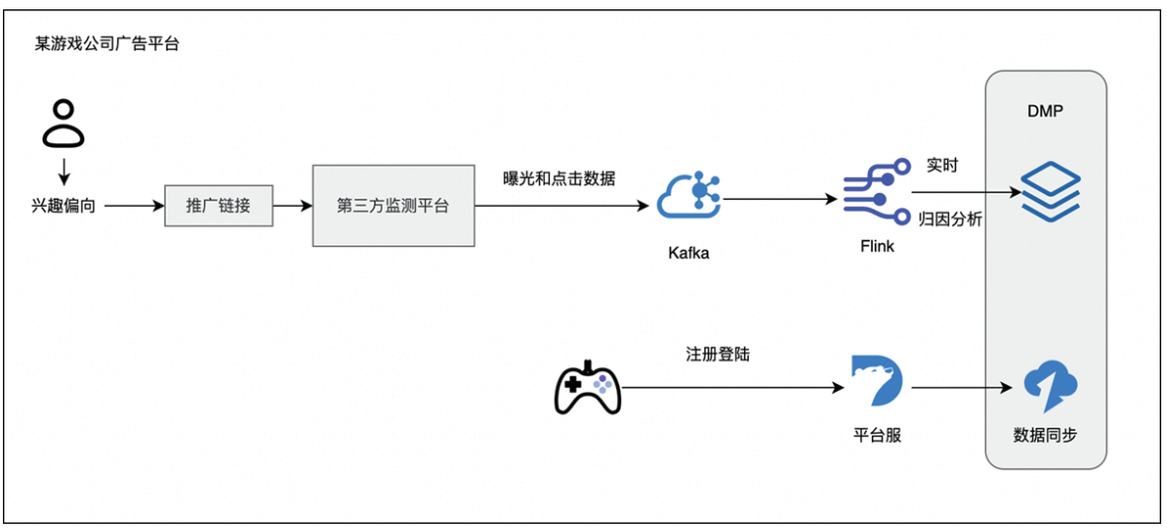

图47

6.2 大数据产品能力及湖仓方案

6.2.1 大数据产品介绍

- Maxcompute

阿里云MaxCompute为游戏行业提供了一套高效、安全的大数据计算与分析解决方案，能够支撑游戏公司应对用户行为分析、运营决策、业务增长等核心场景的挑战。面对每日产生的PB级玩家数据（如登录日志、付费流水、关卡行为、社交互动等），MaxCompute无需预先扩容即可实现弹性扩展，快速完成数据清洗、统计与深度挖掘——无论是实时监控新版本上线后的玩家留存率、精准定位高价值用户的付费特征，还是通过机器学习预测活动效果并提前优化玩法设计，均可通过低代码开发与灵活的资源调度高效落地。同时，其按量计费的特性大幅降低运维成本，免去自建大数据平台的高昂投入，让团队聚焦于数据驱动的精细化运营，从玩家体验优化、商业化策略制定到跨平台用户画像构建，全面释放数据价值，助力游戏业务持续增长。

图48

- 实时计算Flink版

阿里云实时计算Flink版为游戏行业提供了高吞吐、低延迟的流式数据处理能力，支持毫秒级实时响应与复杂事件驱动型场景。基于Flink原生流计算引擎，可无缝对接游戏服务端日志、玩家行为埋点、IoT设备数据等多源流式数据，实现玩家在线状态监测、实时反外挂风控、动态弹窗策略执行等关键业务。例如，通过CEP（复杂事件处理）规则即时识别异常战斗行为并触发封禁逻辑；结合时间窗口聚合统计全服分时段活跃用户，驱动服务器资源弹性调度；依托精准一次（Exactly-Once）语义保障付费流水等核心数据的端到端一致性。同时，其全托管服务架构支持自动扩缩容与故障恢复，显著降低流计算场景的运维复杂度，助力游戏企业构建实时化、智能化的运营体系，快速响应玩家需求并提升业务决策效率。

- Hologress

Hologres是一款实时数仓产品，结合了大数据处理和实时分析的能力，特别适合需要快速响应和高效数据处理的游戏运营场景。在游戏运营中，Hologres可以帮助实时监控用户行为、游戏性能和收入情况，从而支持精准的广告投放、优化运营策略和提升用户体验。通过Hologres，游戏公司可以快速获取多维度数据，进行实时分析和决策，确保游戏的稳定运行和持续增长。

图49

- Starrocks

StarRocks是一款高性能实时分析型数据库，专为大规模数据实时查询和分析设计，能够支持高并发、低延迟的场景需求。在游戏运营场景下，StarRocks可以实时处理用户行为数据、游戏内事件数据和收益数据等，帮助游戏公司快速进行多维分析，如用户活跃度、留存率、付费行为和游戏内经济系统平衡等。通过StarRocks，游戏运营团队可以实时监控游戏表现，快速优化运营策略，提升用户体验和游戏收益。

图50

- QuickBI

Quick BI是一款高效的数据可视化与分析工具，支持多数据源的快速连接、数据建模和可视化展示。在游戏运营场景下，Quick BI可以帮助运营团队实时监控用户活跃度、付费行为、留存率、活动效果等关键指标，通过灵活的仪表盘和多维分析功能，快速生成数据报告，支持精准决策。例如，运营人员可以通过Quick BI分析用户画像、游戏内行为路径和收入分布，优化广告投放策略、调整游戏内容和活动设计，从而提升用户体验和游戏收益。Quick BI的高效性和直观性使其成为游戏运营数据分析的重要工具。

图51

6.2.2 云原生实时湖仓

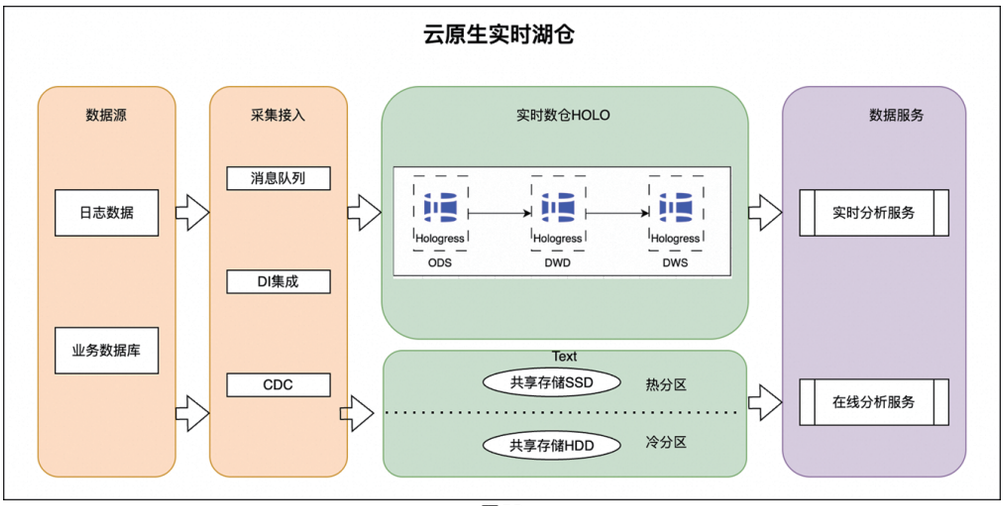

图52

数据源：

- 包括在线业务库等多种数据源对接，也可接入如SLS等非结构化数据

采集接入：

- 通过Kafka等消息队列做数据缓冲区，可以做消息回放处理，通过Dataworks DI集成从Kafka全量/增量同步数据
- 基于Flink cdc实现非/半结构化数据同步，支持CDAS/CTAS等语法配置支持全/增量同步，整库同步，分库分表同步等场景
- 实时湖仓Hologres，每一层数据都支持高效更新与修正、写入即可查，解决了传统实时数仓解决方案的中间层数据不易查、不易更新、不易修正的问题。实时ETL链路基于Flink SQL实现，分为常见的ods，dwd，和dws层，统一存储在数据湖中，架构简单，模型统一。Hologres支持通过主从实例读写分离部署（共享存储）或计算组实例架构实现资源强隔离，从而保证Flink对Hologres Binlog的数据拉取不影响线上服务。

方案优势：

- 性能优势
  - 结合 Flink 的流处理能力和 Hologres 的高性能查询与写入功能，支持即时数据分析。Hologres 能够作为高效的维表进行快速查询，同时保证高吞吐量的数据更新和查询速度，实现数据的实时分析和服务。
- 简化技术栈与成本
  - 通过使用Flink cdc + Hologres 减少了对多种开源组件（比如kafka）的依赖，通过简化数据处理链路，降低了系统复杂度和技术学习成本。
- 灵活的数据集成与优化的数据处理流程
  - 支持流批一体和湖仓一体的数据处理模式，兼容 Delta、Hudi 等格式，以及 PostgreSQL 生态，使得现有应用迁移变得简单。它还支持局部更新和 Binlog 实时通知，简化了多流 join 的复杂度，提升了数据处理效率。

6.2.3 开源生态实时湖仓

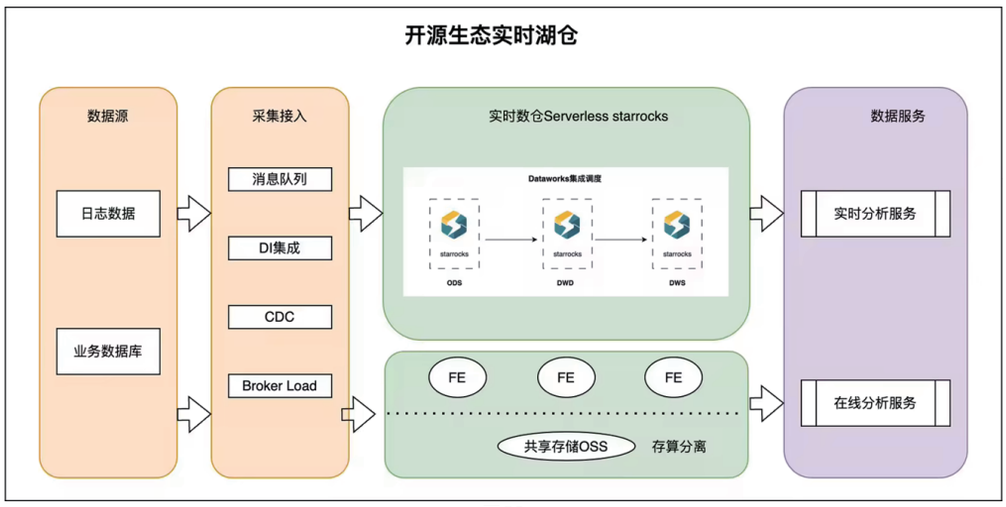

图53

数据源：

- 包括在线业务库等多种数据源对接，也可接入如SLS等非结构化数据

采集接入：

- 除常规的Flink cdc外，可以通过Dataworks数据集成实现数据的全/增量同步，如果需要做纬表join实时计算，还需要维护Flink实时流作业
- 数据湖中如事件类数据可以通过Hive做统一处理引擎做数据加工，而后通过StarRocks的broker load完成数据同步

方案优势：

- 高性能的实时数据分析：
  - StarRocks 支持高效的实时数据写入和查询，能够在秒级延迟内处理大规模的数据流。其分布式架构设计允许线性扩展，以应对不断增长的数据量和查询负载。此外，StarRocks 采用向量化执行引擎优化了查询性能，使得复杂的分析任务也能快速完成。
- 简化数据架构与降低运维成本：
  - StarRocks 提供了一站式的实时数仓解决方案，减少了对多种不同技术栈的需求，从而简化了整体数据架构。用户可以使用 SQL 直接操作数据，无需额外的ETL过程，降低了开发和维护的成本。同时，StarRocks 的自动化运维功能，如自动分区、自动复制等，进一步减轻了运维负担。
- 强大的兼容性和灵活的部署选项：
  - StarRocks 兼容 MySQL 协议，支持标准的 SQL 查询语言，使得现有应用迁移变得简单快捷。它还支持多种数据源的接入，包括 Kafka、HDFS 等，方便用户整合不同的数据来源。此外，StarRocks 支持云原生部署以及私有化部署，可以根据企业的具体需求选择最适合的部署方式

6.3 常见问题及运维保障

6.3.1 实时计算-Flink的高频问题

- 数据延迟

  Flink中的数据延迟是一个比较常见的问题，从发生延迟的位置来看，可以分为：

  - source端的数据延迟；
  - Flink处理过程中的数据延迟；
  - sink端的数据延迟（sink慢，产生背压）。

在游戏行业中，比如在针对广告场景中实时竞价的场景中，需要根据外部数据做大量的计算分析，在游戏公司精细化运营的背景下，甚至需要匹配单个的用户画像来做竞价分析，从数据处理的链路上来看，这时Flink负载会加大，如果出现部分算子的性能瓶颈，就很可能造成数据延迟，进而产生业务受损的情况。针对这一常见问题，有如下的排查思路：

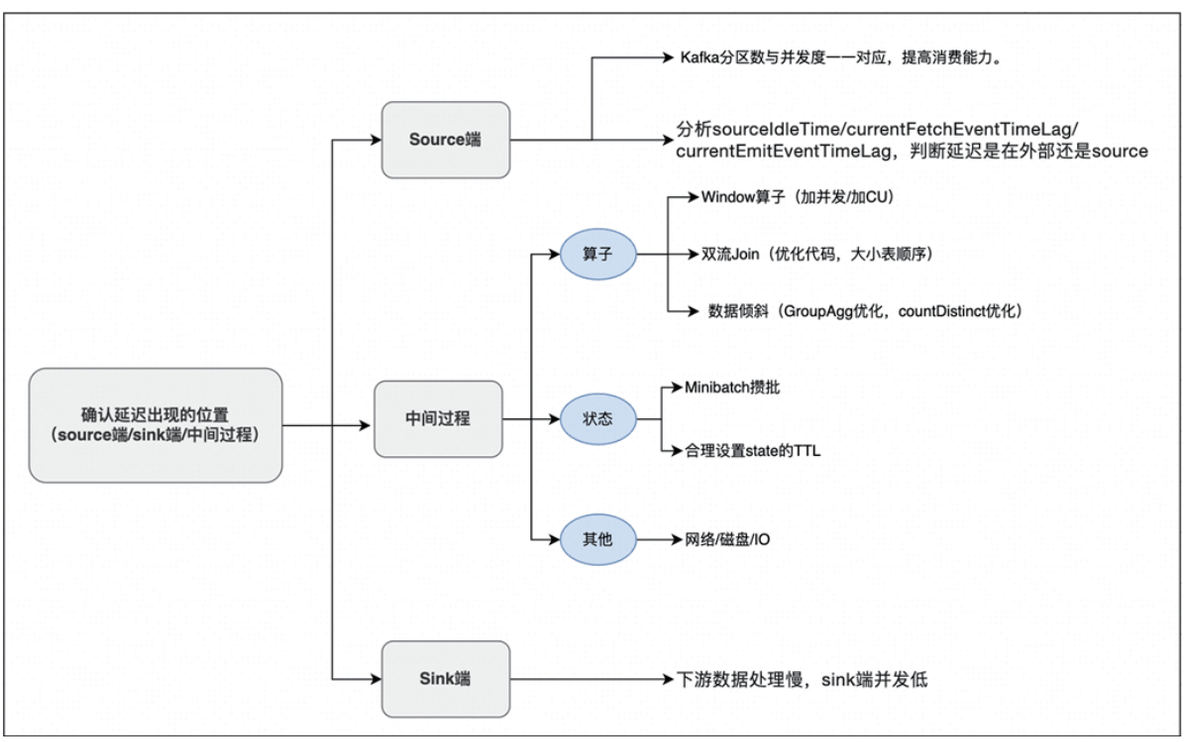

图54

这里列出了一些常见的问题排查思路及手段，实际问题中还可以利用火焰图，堆栈日志等手段来定位分析问题。

- 数据处理-Flink大状态调优

通过广告获取用户后，需要对用户行为做分析，然后再通过结果来进一步优化广告投放，进而提升获客的效率，Flink用来获取用户点击日志，再去查StarRocks中的广告数据，在Flink上做广告的归因分析，这里就会涉及到大状态算子。针对大状态的调优可以岸如下的思路进行

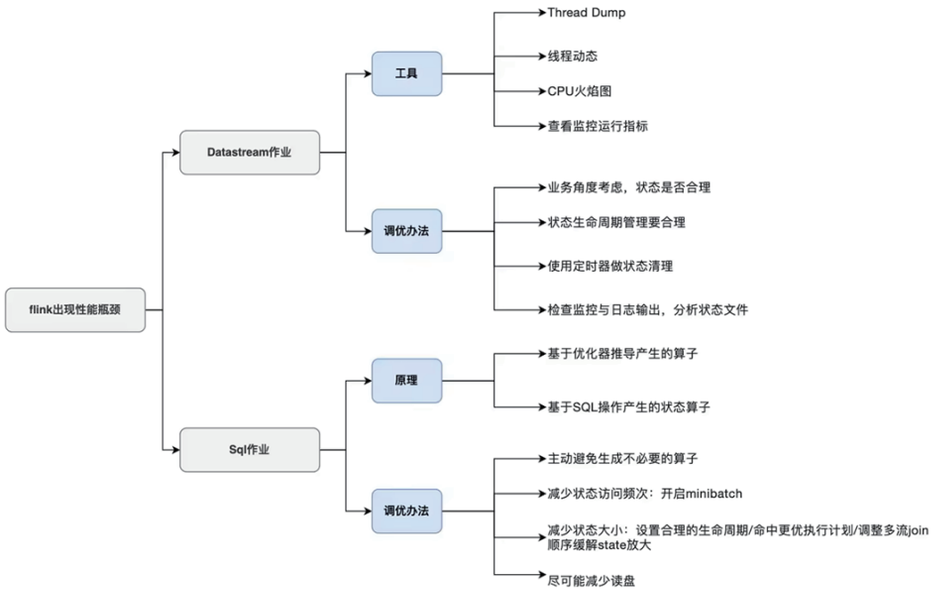

图55

6.3.2 数据备份问题

- MaxCompute备份与恢复

MaxCompute提供数据备份与恢复功能，系统会自动备份数据的历史版本（例如被删除或修改前的数据）并保留一定时间，您可以对保留周期内的数据进行快速恢复，避免因误操作丢失数据。该功能具有以下几个特点：

- 默认开启，不需要手动开通：该功能不依赖外部存储，系统默认为所有MaxCompute项目开放的数据保留周期为24小时，备份和存储免费。
- 自动持续备份：系统自动对发生变更的数据进行备份，多次变更时将备份多个数据版本，相比固定周期性的备份策略，可以有效避免因误操作丢失数据。
- 恢复快速，操作简单：MaxCompute具备先进的元数据和多数据版本管理能力，备份和恢复操作不占用额外的计算资源，您可以通过命令快速恢复不同规模的数据。

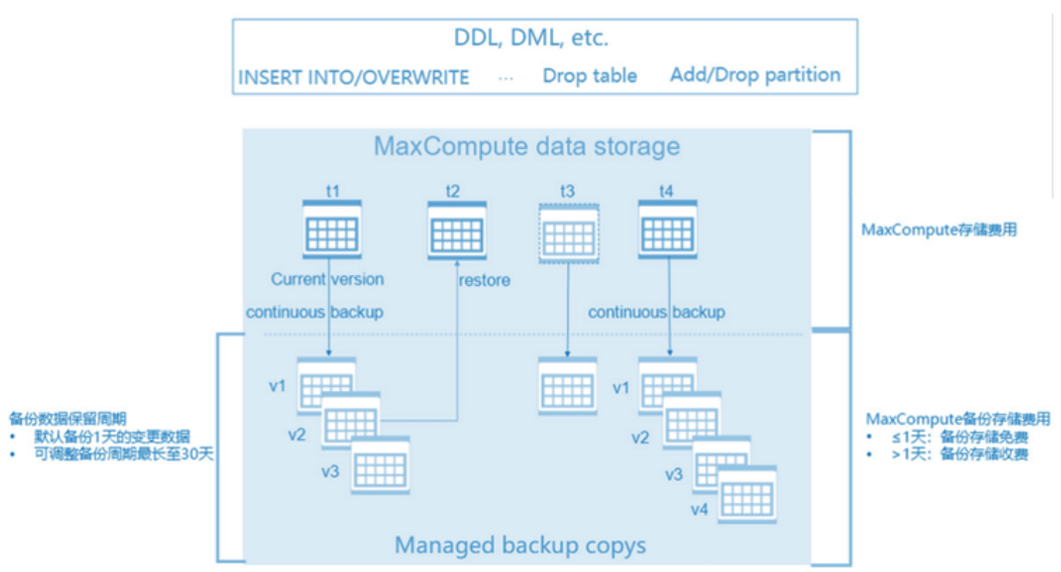

图56

- Hologres的备份与恢复

Hologres支持自动周期备份和手动备份，主要用于应对误操作（如数据删除或更新）以及灾难恢复场景。通过备份功能，用户可以将实例的历史数据保存下来，并在需要时恢复到新实例。该功能有以下几个特点：

- 自动周期备份：用户可以设置备份周期和时间，系统会按照设定的时间自动生成备份快照。
- 手动备份：用户可以根据需求随时触发手动备份任务。
- 备份存储：存放备份的地域是Hologres实例所在的地域，支持跨可用区备份和恢复，仅部分Region支持跨Region备份与恢复。

6.3.3 产品容灾能力

- 大数据开发治理平台Dataworks

DataWorks 在管控面支持多区域部署（管控自带多AZ高可用，DB/OSS三备份，对用户透明），计算端不支持自动容灾。当灾难发生时，资源组逃逸需依赖客户自行新购入的资源，产品侧会根据资源组形态（ECS、ACK）自动调度到其他可用区从而实现逃逸。

- 实时计算Flink

Flink在管控面默认支持多AZ部署；在计算面从产品形态上看，对于按量付费形态支持单可用区，该形态Flink任务失败后通过重启自动逃逸到底层正常集群上，但容灾场景下Flink任务受到用户上下游数据源和OSS的影响，依赖用户迁移；对于采取包年包月付费形态，支持开启跨可用区功能来实现多AZ部署（同城高可用），使用跨可用CU，当主可用区出现故障后，作业将在用户选择的备可用区恢复。

- 实时数仓Hologress

Hologres管控层面默认多AZ部署，但计算及存储面为单可用区实例，底层依赖物理机不具备机房级别故障的逃逸能力。但Hologres支持多可用区容灾部署，在一个region的2个不同的可用区创建实例，基于主实例和灾备实例直接复制数据，实现跨 AZ 容灾。Hologres对外使用一个SLB（Endpoint）读写。默认情况下Endpoint指向主机房，数据从主机房向容灾机房的实例同步。当主机房发生故障时，需要用户主动进行切换， SLB（Endpoint）指向容灾实例。在主机房故障排除后，会自动恢复备机房向主机房的数据备份。其中，主实例和灾备实例的计算组，需要 1:1 的计算和 1:1 的存储。

6.4 未来展望

随着流批一体，湖仓一体，Data+AI发展趋势，未来，阿里云大数据产品在游戏行业中的运营场景将更加实时化，全球化，智能化。通过深度融合大数据+AI、云游戏等新技术，阿里云将帮助游戏公司实现数据驱动的精细化运营，提升玩家体验和商业价值，随着技术的不断进步和行业生态的完善，大数据将成为游戏行业创新的核心驱动力之一。

第七章 游戏美术：构建幻想世界的画笔与色彩

7.1 什么是游戏美术

游戏美术，简而言之，是指为电子游戏创作视觉元素的艺术过程。它涵盖了从概念设计到最终成品呈现的所有视觉创作环节，包括但不限于角色设计、场景构建、界面UI/UX设计、特效制作、动画制作以及后期合成等。游戏美术师们通过精湛的技艺和无限的创意，为玩家构建出一个既真实又充满想象力的游戏世界。

7.2 游戏美术的设计阶段

7.2.1 角色设计

角色设计是游戏美术的核心之一。每个游戏角色都是其背后故事的载体，是玩家情感的寄托。游戏美术师们需要根据游戏的世界观、剧情设定以及角色性格，设计出各具特色的角色形象。这包括角色的外貌特征、服装风格、武器装备乃至表情动作等每一个细节。优秀的角色设计能够迅速抓住玩家的眼球，引发共鸣，成为游戏中的明星角色。

图57 ((图片由通义万相生产，仅做示意))

图58 (图片由通义万相生产，仅做示意)

7.2.2 场景构建

场景是游戏世界的舞台，是玩家探索与冒险的空间。游戏美术师们通过精细的场景构建，营造出或壮丽、或神秘、或温馨、或恐怖的游戏环境。他们运用色彩搭配、光影效果、材质纹理等多种技术手段，将平面的设计稿转化为立体的游戏场景。这些场景不仅要美观，更要符合游戏的逻辑与氛围，为玩家提供沉浸式的游戏体验。

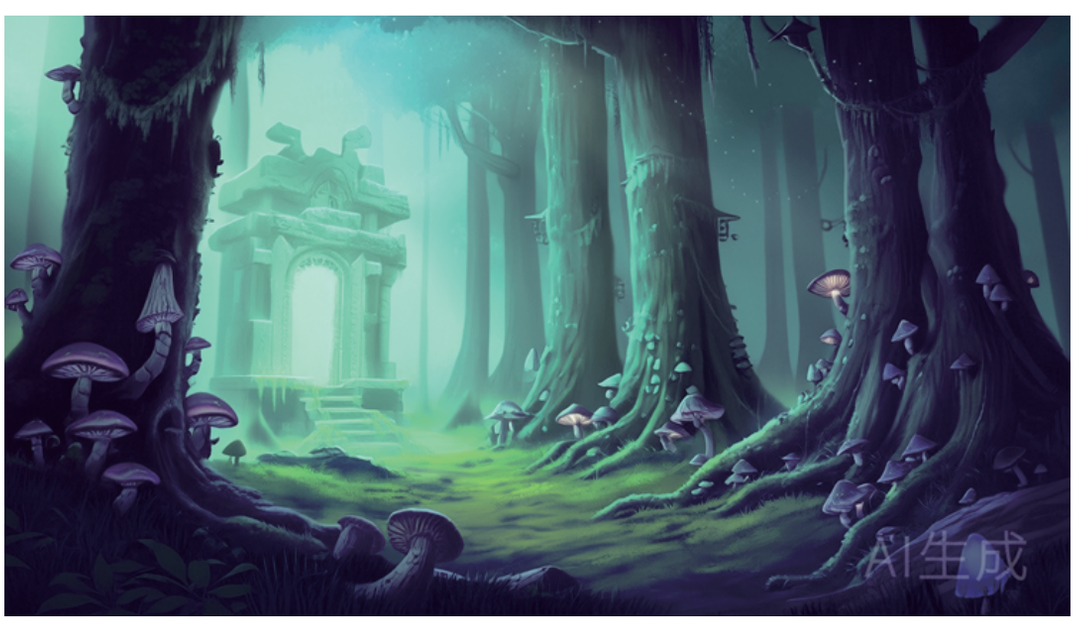

图59 （图片由通义万相生成，仅做示意）

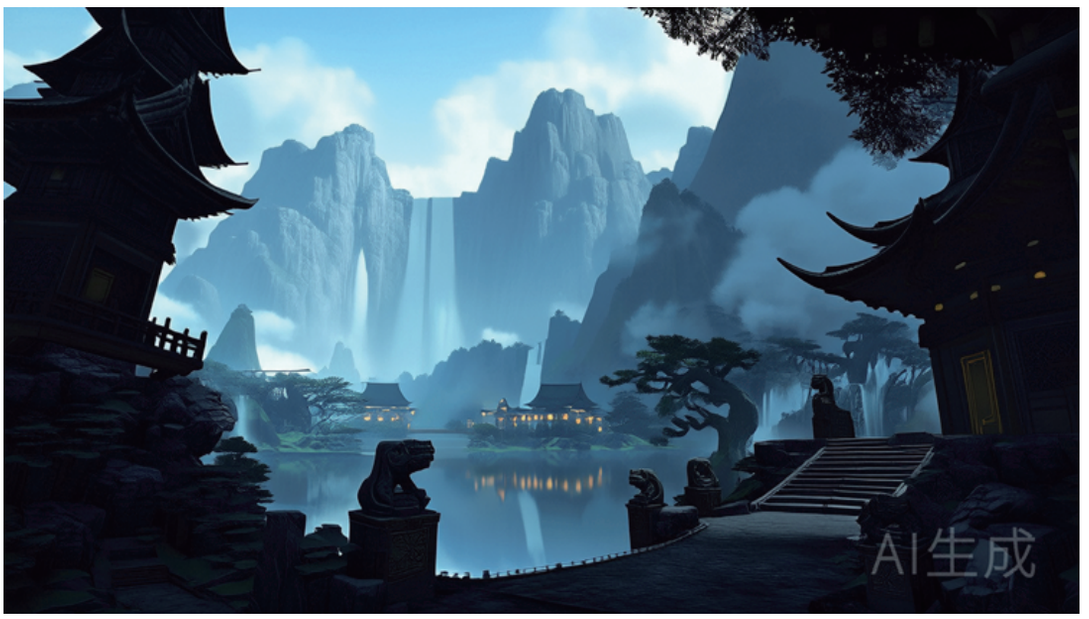

图60 （图片由通义万相生成，仅做示意）

7.2.3 界面UI/UX设计

界面UI/UX设计是游戏美术中不可或缺的一环。它直接关系到玩家与游戏之间的交互体验。优秀的UI设计能够清晰地传达游戏信息，引导玩家操作；而UX设计则注重提升玩家的使用感受，让游戏操作更加流畅自然。游戏美术师们需要不断优化界面布局、图标设计、色彩搭配等，确保玩家在游戏中能够享受到便捷、舒适的交互体验。

图61 （图片由通义万相生成，仅做示意）

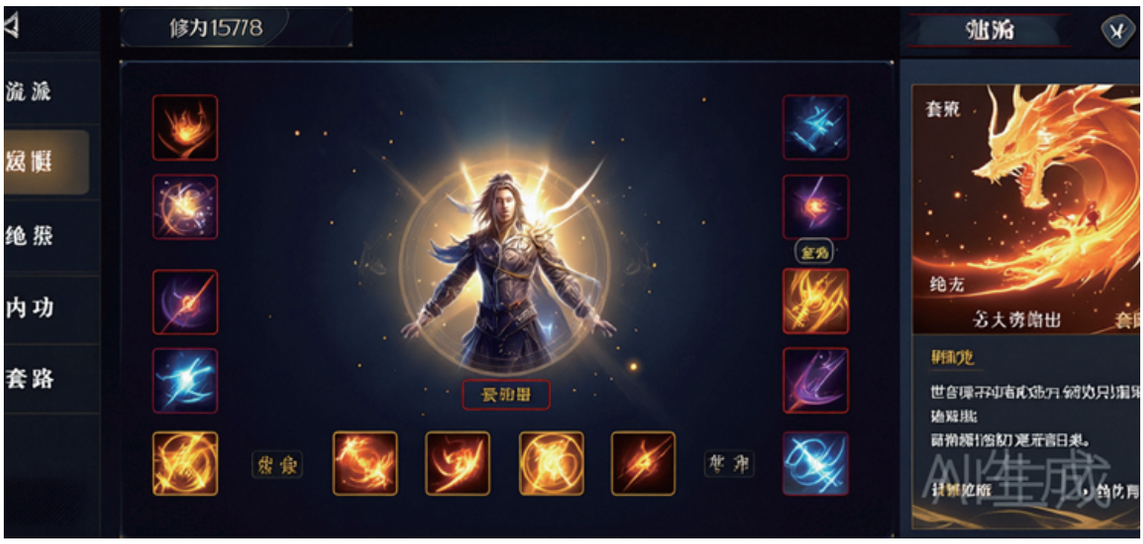

图62 （图片由通义万相生成，仅做示意）

7.2.4 特效与动画制作

特效与动画是游戏美术中的点睛之笔。它们能够赋予游戏画面以生命力，让静态的场景和角色变得生动起来。特效制作包括爆炸、火焰、水流等自然现象的模拟，以及技能释放、打击感等游戏元素的呈现。而动画制作则涉及角色动作、场景过渡、剧情演绎等多个方面。游戏美术师们通过精心的设计与制作，让游戏中的每一个瞬间都充满惊喜与动感。

图63 （图片由通义万相生成，仅做示意）

图64 （图片由通义万相生成，仅做示意）

7.2.5 后期合成与优化

在游戏美术的最后阶段，后期合成与优化工作显得尤为重要。这一环节主要负责对游戏画面进行整体调整与优化，确保游戏在不同平台、不同设备上的显示效果都能达到最佳状态。游戏美术师们会运用各种后期处理技术，如色彩校正、光影调整、细节增强等，对游戏画面进行精细打磨。同时，他们还会对游戏性能进行优化，确保游戏在保持高画质的同时，能够流畅运行。

图65 （图片由通义万相生成，仅做示意）

图66 （图片由通义万相生成，仅做示意）

7.3 游戏美术创作中的痛点

7.3.1 跨部门沟通与信息碎片化

需求变更频繁，关键参数（如色域标准、分辨率、帧率等）需要反复确认，30%的创作时间被无效沟通消耗，导致效率低下。

7.3.2 基础能力不足

新手容易忽视人体结构的比例，过于关注细节（如肌肉肌理），而忽略了整体型体。知道理论但无法将其应用到实际设计中，缺乏对“型”（外轮廓/剪影）和“体”（体积）的理解。

7.3.3 创作与效率的平衡

开发周期延长，尤其是在独立游戏团队中，资源有限的情况下更显突出，难以快速满足多样化需求。

7.4 AIGC在游戏美术中的局限与挑战

7.4.1 创造力的局限

AIGC本质上是基于已有数据进行训练，其生成的内容更多是对已有风格的重组或变形，缺乏真正的原创性。这意味着它难以替代人类艺术家在创造全新风格或概念方面的能力。

7.4.2 艺术质量的控制

AIGC生成的内容质量参差不齐，特别是在复杂项目中，可能需要设计师对AI输出结果进行大量的后期调整和优化。

7.4.3 版权问题

AIGC的训练数据通常来自公开网络，而这些数据可能涉及版权争议。在游戏开发中，如何避免侵犯他人知识产权是一个需要解决的重要问题。

7.5 AI能带给游戏什么

7.5.1 AI在游戏美术场景使用的优势

- 快速生成概念设计，AIGC通过输入简单的关键词或描述，AI可以快速生成大量风格迥异的视觉稿件，协助开发团队进行脑暴
- 游戏场景的生成与优化，AIGC通过生成程序化的场景素材，大幅减少了设计师的工作量

7.5.2 AI在游戏美术场景使用的弊端

- 通义万相创作的画作，很大概率无法满足创作者的主观诉求
- 通义万相创作的画作，有一定概率会触及版权问题
- PAI应用于图生图的场景时，受限较多，即便训练了大量的原版草图，在生成符合游戏策划人物以及场景的图时，达到60分的草图占比还是太低

7.5.3 通过AI，能在游戏行业“实际”能做些什么？

- 游族网络将AI应用于美术资产生产、本地化多语言版本制作、音频制作、质量管理、NPC等模块，提升内容生产制作的效率
- 腾讯在GDC上则发布了自研游戏AI引擎GiiNEX，展示GiiNEX在3D城市生成和UGC关卡设计场景中的作用
- 天美F1工作室基于机器学习自研的角色动画系统，能在位移动画和交互动画两方面优化角色的动作质量，让两种动画更真实、生动
- 腾讯互娱魔方《暗区突围》手游项目，则开发了一套混合渲染管线，首次在移动端场景中应用了光线追踪技术，使游戏画面更真实
- 暴雪的《守望先锋》与腾讯的《王者荣耀》中，都引入了AI bot技术，让人机对战中的机器人对手具有更贴近真人的操作习惯，从而加强玩家在人机对战中的实际练习价值

7.5.4 AI生成内容引领游戏变革

1. 大型模型多模态进化：游戏创新加速

未来AI可以直接输出不同模态的游戏内容，如3D视觉内容、音频内容等，而不需要游戏制作者自己花时间逐项制作。这可以极大地缩短游戏的制作周期，降低内容创作的难度。同时，AI强大的创意能力也将成为游戏开发者的脑力资源，输出独特的游戏机制、剧情等创意内容。

在游戏体验上，AI生成的沉浸式视听效果将使游戏向电影级的多模态交互艺术转型。依托AI产出丰富的游戏素材，游戏的艺术表现力也将由此达到新的高度。

2. AI 3D工具：提升游戏资产制作效率

AI 3D工具通过应用新技术，正在改变和提高3D资产的制作流程，从而大幅提升3D内容的制作效率。

具体来说，利用神经网络技术，只需要输入文字描述，AI就可以自动生成3D模型；通过人脸识别技术，使用极少的图片数据就可以快速还原高保真的3D人脸；利用生成技术，可以自动构造复杂的3D场景和建筑，而不再需要手工制作。

主流游戏引擎也在探索使用AI辅助开发3D游戏内容。未来利用AI生成3D资产，将大幅减少重复劳动，使创作者可以将更多时间放在内容的创新上。

3. AI提升游戏体验品质

AI智能可以设计出更人性化的游戏NPC，与玩家进行情感交流，让游戏世界更具沉浸感。同时，可以训练出不断进步的机器人对手，根据不同玩家的水平提供个性化的挑战，使游戏对战更有趣。

AI智能可以通过分析玩家的游戏数据和习惯，比如档案资料、浏览记录等，绘制出精准的用户画像。在此基础上，利用强化学习不断优化算法，主动给玩家推荐感兴趣的游戏。同时，可以分析每个玩家的喜好和游戏水平，然后自动匹配一个合适的游戏。让玩家可以尽兴地玩游戏。

7.6 通义系列带给游戏美术的价值

7.6.1 极大地提升美术创作效率

快速生成高质量素材

通义万相支持通过文字描述快速生成符合需求的图像（如角色、场景、道具等）供画师参考，甚至支持风格化调整（如3D卡通、油画、水彩等）。例如，设计师只需输入“赛博朋克风格的未来城市”，即可快速生成初步设计，大幅减少从零开始绘制的时间。

降低重复性劳动

巨人网络通过与PAI合作开发的AI绘画平台iMagine，将角色和场景原画的制作效率提升了50%-70%，UI和图标设计效率甚至提升了80%以上。AI可快速生成基础素材，让美术团队更专注于创意优化和细节调整。

7.6.2 创意辅助与灵感激发

无限风格可能性

通义万相和PAI的AI绘画工具能够生成多种艺术风格的图像（如写实、抽象、像素风等），帮助设计师突破传统创作限制，快速尝试不同视觉方案。例如，游戏开发者可以轻松切换“梦幻童话风”或“赛博朋克风”来测试市场反馈。

动态视觉效果生成

通过与游戏引擎结合，AI可实时生成动态视觉元素（如光影、粒子效果、环境变化）。例如，根据玩家行为或剧情进展，AI可动态调整场景的雾效、水面波动或法术特效，增强沉浸感。

7.6.3 个性化与玩家共创

玩家定制化体验

AI绘画技术允许玩家通过简单交互（如输入文字描述或调整参数）生成个性化角色、装备或关卡。例如，玩家可设计“一只穿着西装打领带的猫”作为游戏角色，AI快速生成并允许调整颜色、背景等细节，提升玩家参与感和粘性。

UGC（用户生成内容）支持

游戏可集成AI工具，让玩家自主创作内容（如皮肤、地图），形成社区共创生态，延长游戏生命周期。

7.7 总结

AIGC在游戏美术设计中的作用已从辅助工具逐渐转变为重要的创作力量，AIGC虽然无法替代艺术创作，但是它可以学习更多的风格元素，不断迭代，为初版原画创作提升创作效率。不过这一切基于语言理解算法的迭代，使其能够充分理解文字。

随着技术的不断发展，相信AIGC与人类设计师的协同创作模式将成为游戏美术设计的新常态，为玩家带来更丰富、更创新的视觉体验，为所有游戏玩家带来不一样的角色视觉冲击。

第八章 游戏内容审核：智能守护虚拟世界的多元表达

在数字内容爆炸式增长的时代，内容审核已成为维护游戏平台健康生态系统的关键。在游戏行业中，发言、头像等玩家表达容易因为玩家猎奇、跟风、恶意或非主观恶意产生违规内容，这种违规内容带来的后果可大可小，因此游戏产商在内容审核会投入较大的成本。同时从2024年起，游戏行业持续创新，玩家创作形式从平面图像扩展到3D模型、视频流、实时语音。以《原神·尘歌壶》家园系统为例，每件玩家作品都生成多张俯视图和社区截图；同时，社交直播弹幕、跨服联动的语音互动，也对审核提出了更高要求。为保证社区生态与品牌信誉，游戏厂商对内容审核系统提出了四大核心指标：

- **日均审核请求量：**UGC玩法上线后，一天内审核请求可达数万至数十万；
- **审核延迟（RT）：**从玩家提交到反馈需控制在1-5秒级，保证交互流畅；
- **审核准确度：**不仅要高召回（Recall），更要兼顾高精准（Precision），避免对合法内容误判；
- **人力成本：**传统人工审核团队规模需动态弹性，应对热度波动却不能过度浪费资源。

这些KPI直接关系到玩家体验与运营成本，必须在保证合规安全的前提下，最大化自动化与智能化水平。

8.1 游戏内容审核背景

8.1.1 内容审核的定义与重要性

内容审核是游戏平台中的内容符合其既定政策和社区标准而进行审查、过滤和管理的过程。这涵盖了用户在游戏中的聊天文字、游戏中上传的图片头像、游戏中用户生产的内容等。内容审核的重要性体现在多个维度：

首先，它对于确保安全与隐私至关重要。审核旨在移除可能对个人人身安全造成风险的内容，保护个人隐私信息，并清除非法或滥用性内容。对于儿童等弱势群体而言，内容审核尤为关键，因为它能过滤掉儿童性虐待图像等非法内容，以及可能有害但不一定违法的内容，例如宣扬饮食失调的材料。

其次，内容审核通过维护一个安全的环境来支持言论自由。当用户感到安全时，他们更愿意表达自己的观点。如果游戏平台不进行审核，有害内容将泛滥，可能导致用户流失并扼杀健康的交流。因此，审核作为一种必要的“守门”功能，有助于维护公共领域的整体健康，而非简单地限制言论。这种动态关系表明，有效的内容审核不仅是技术或法律合规问题，更是促进健康、活跃和可持续在线社区及民主话语的基础。它将审核从“限制”转变为在既定安全边界内“赋能”表达的手段。

最后，内容审核是建立信任、促进收入与增长的必要商业投资。有效的审核能够增强用户信任，提升品牌声誉，对于依赖订阅服务或广告商的平台而言，是吸引和留住用户群及收入的关键。它有助于培养积极的在线文化，并维护在线平台的完整性。

8.1.2 数字时代内容激增带来的挑战

用户生成内容（UGC）的爆炸式增长，据估计每天产生约403 EB的数据，使得传统的人工内容审核变得力不从心且不可持续。这种海量的数据加剧了检测和移除不良内容的难度，也使得平衡言论自由与监管、确保审核一致性成为巨大挑战。有害内容和虚假信息的迅速传播进一步凸显了对实时、可扩展审核解决方案的迫切需求。

UGC的巨大体量不仅是一个挑战，更是推动AI应用的核心驱动力。人工审核在如此规模下几乎“不可能”实现，且“不足以”应对。这种内在的规模要求直接导致了从人工审核向AI驱动或人机混合审核模式的转变。内容在几分钟内就能病毒式传播的速度进一步强调了自动化、实时响应的必要性。这表明AI在内容审核中的应用并非仅仅为了提高效率，而是数字平台在现代数字环境中生存和运作的根本需求。没有AI，有害内容的规模可能会使平台无法使用或在法律上无法维持，甚至导致游戏下架。

8.1.3 AI在内容审核中的作用演变

最初，内容审核主要依赖人工和小模型NLP关键字提取等方式进行。然而，随着内容量的激增，游戏平台开始引入AI工具来处理海量信息。AI系统现在已成为“第一道防线”，能够以极快的速度筛选大量数据，进行事实核查和内容过滤。它们甚至可以在内容被用户举报之前就检测并移除有害信息。

AI在内容审核中的作用已从简单的关键词过滤演变为更高级的语言分析、图像识别和语境理解。这种演变不仅仅是自动化，更是对人类能力的增强。AI在自动化初始过滤和处理高流量任务的同时，也通过将“可疑内容转交人类内容审核团队”来增强人类审核员的能力，使他们能够“专注于更复杂和细致的审核任务”。这标志着从简单自动化向更复杂伙伴关系的转变。其目标是“减少人工工作量”并“将人工审核成本降低高达90%”，而非完全取代人类。这预示着未来人类和AI的角色将日益专业化和互补，AI处理审核的“日常工作”，而人类则专注于“边缘案例”和复杂的语境判断。这种混合模式 正在成为事实上的标准。

8.2 游戏内容审核的场景与类型

8.2.1 主要应用场景

在现代网络游戏，尤其是大型多人在线游戏（MMO）、社交游戏和用户生成内容（UGC）平台中，玩家的互动日益频繁，表达形式也愈发多样。然而，这种开放性也带来了内容安全的风险，如不当言论、违规图片、恶意信息传播等，不仅影响玩家体验，还可能触碰法律法规红线，甚至引发社会争议。因此，构建高效、智能、全面的内容审核体系，已成为游戏运营不可或缺的一环。

游戏社区媒体平台

游戏社交媒体平台（如taptap平台，各类游戏论坛网站）是用户分享游戏玩法、游戏角色和参与游戏内容讨论的主要渠道，同时也是不良信息传播的重要载体。这些平台每天需处理海量的评论、私信、图片。内容审核是保障社区安全、维护舆论环境、执行平台政策的重

要手段。头部平台通常结合机器审核与人工复核，使用多模态模型实现涉政、谣言、低俗、暴力等多类别检测。

在线游戏与聊天服务

游戏和语音聊天场景流量高、互动实时，用户生成内容包括世界频道、公会聊天、私聊、队伍语音转文字、表情包文字描述、语音、游戏昵称、动态等，需要严格监控。此类环境面临辱骂、歧视、色情、广告引流、政治敏感、仇恨言论、欺凌等行为的普遍存在，在部分玩家文化中甚至被视为“默认”。网易、莉莉丝、米哈游等公司在游戏场景中均投入大规模内容审核系统，能够理解本地俚语、网络用语和语音识别结果，支持实时风控和高并发处理。

游戏头像与图片分享

玩家自定义头像、角色形象截图、公会徽章、个人主页背景图等图像内容，是个性化表达的重要方式。但部分用户可能上传包含色情、暴力、血腥、敏感符号、政治人物或品牌侵权的图片。也可能会有使用知名动漫、影视角色或商标作为头像。甚至通过对抗性伪造，通过滤镜、拼接、模糊处理规避识别。这种场景需要识别图像中的敏感元素，识别图像中的文字信息结合文本审核，利用多模态大模型进行细粒度语义理解，判断图像是否具有冒犯性或隐含违规意图。

游戏用户生成内容(UGC)

随着游戏创作生态的发展，越来越多游戏支持玩家生成内容，如《我的世界》的地图、《Roblox》的游戏关卡、《原神》的社区创作、《永劫无间》的皮肤设计等。这些UGC内容形式丰富，审核复杂度高。游戏中地图/场景设计可能包含色情场景、反社会主题（如监狱、刑场）、政治隐喻建筑。用户创作文本如任务描述、剧情对话、物品命名中可能夹带违规信息。这种场景需要分别提取文本、图像、音频特征，进行联合分析。利用多模态大模型对整体内容进行“意图评估”和“氛围判断”，例如识别一个地图是否营造出“恐怖压迫”或“色情暗示”的氛围。针对已发布内容进行动态监控，发现问题可快速下架。

游戏直播审核

随着游戏直播行业的迅速发展，越来越多的玩家和观众通过直播平台观看游戏过程、学习游戏技巧、参与互动。然而，这种开放性和实时性的交流形式也带来了内容安全的新挑战。游戏直播审核旨在确保直播内容健康、合法、符合社会道德规范，同时维护良好的用户体验。直播过程中展示的游戏画面是主要内容之一，但有时可能会出现不适宜的内容，如暴力血腥场景、色情暗示、敏感政治符号等。部分主播可能会利用游戏画面进行不当行为演示，例如作弊或使用外挂。弹幕中可能出现辱骂、骚扰或其他不适合公开讨论的话题。这种场景实施自动化审核机制，结合人工复审以处理复杂情况，同时也可以根据不同的观众群体和地区设定相应的审核标准。

8.2.2 需审核的内容类型

内容审核所针对的有害内容类型广泛且不断演变，需要细致入微的识别和处理。

表19

类别	描述	示例
非法内容	违反法律法规的内容，如儿童性虐待材料、恐怖主义宣传。	儿童性虐待图像 (CSAM)；宣扬恐怖主义的材料
仇恨言论与骚扰	针对个人或群体的贬低、攻击性或煽动性言论，可能导致他人不适或退出讨论。	“你这个废物”；种族歧视性称谓；性别歧视言论
虚假信息与误导性内容	虚假、不准确或旨在欺骗的信息，包括假新闻、阴谋论。	关于公共健康问题的虚假报道；政治阴谋论
成人与露骨内容	包含裸露、性暗示或露骨描绘的文本、图像或视频	裸体图片；性暗示文本；露骨视频
垃圾信息与诈骗	未经请求的、重复的或旨在欺骗用户以获取利益的内容。	垃圾邮件；钓鱼链接；虚假广告
自我伤害与暴力内容	宣扬、鼓励或描绘自残、自杀或身体暴力行为的内容。	自残指南；血腥暴力图像；威胁性言论
其他政策违规	平台根据自身社区准则定义的其他不当内容。	个人身份信息 (PII)；毒品或武器交易；侮辱性俚语；规避性表达

各类有害内容的不断涌现和演变，构成了**“有害”内容格局的持续演变。这些内容并非一成不变，新的形式不断出现，而“有害”的定义也可能因语境和文化敏感性而异。例如，在游戏环境中，种族/文化仇恨言论常被认为是“非严重”的，这暗示了其常态化。这意味着内容审核是一个不断变化的目标，需要持续的适应。这种动态性要求需要持续研究、采用灵活的AI模型以及强大的人工监督，以跟上不断演变的在线行为和社会规范。这也凸显了制定适用于全球不同语境的通用内容政策的挑战。

8.3 内容审核的方法与技术

内容审核的方法和技术已从纯粹的人工干预发展到高度自动化的AI系统，并最终趋向于人机协作的混合模式。

8.3.1 人工审核

人工审核涉及人类团队审查被标记内容，监控活动并执行政策 。尽管人工审核对于处理细致入微的决策至关重要，但人类审核员面临着可扩展性限制、主观判断导致的不一致性，以及长期暴露于有害内容所带来的巨大心理困扰。

8.3.2 自动化审核

自动化审核依赖算法和人工智能来分析和过滤内容，提供无与伦比的速度、可扩展性和全天候执行能力。

基于规则的系统

这些模型利用高级规则和模式匹配来检测不需要的内容。它们可以检测特定的词语、短语或模式，包括通过重复字符、替换或插入来规避过滤器的尝试。虽然对于明确的违规行为有效，但它们难以处理细微差别、不断演变的语言，并且需要手动提取特征。

机器学习与深度学习模型

这些模型通过学习大量数据集中的模式来识别有害内容，从而随着时间的推移提高准确性和效率。

- **传统机器学习：**早期的模型，如朴素贝叶斯（Naive Bayes）、支持向量机（SVM）和逻辑回归（Logistic Regression），为文本分类奠定了基础 。它们在处理大型词汇表（朴素贝叶斯）或将文本映射到高维空间（SVM）方面表现出色 。特征工程对于这些模型的性能至关重要 。
- **深度学习架构：**近年来，深度神经网络，如卷积神经网络（CNNs）、循环神经网络（RNNs）、长短期记忆网络（LSTMs）和门控循环单元（GRUs），通过捕捉复杂的语义关系和处理序列数据，显著提高了性能。CNNs在处理长文本方面效率高，而LSTMs/GRUs则克服了长序列中的梯度消失问题。
- **Transformer模型：**作为自然语言处理（NLP）领域的最新突破，基于Transformer的模型，如BERT、RoBERTa、GPT和mT5，利用自注意力机制并行处理整个序列，从而捕捉长距离语境并理解细微差别。它们在多语言语境中表现出强大的性能和泛化能力。

自然语言处理技术

NLP技术是AI内容审核的核心，使其能够理解和解释人类语言的复杂性。

- **文本嵌入（Text Embeddings）：** 文本嵌入将非结构化文本数据转换为数值向量，捕捉文本的语义含义。这些嵌入允许系统通过比较向量的相似性来识别语义相似的内容，即使措辞不同，也能检测出有害信息的变体。这对于内容审核至关重要，因为它能识别出使用同义词、拼写错误或重新排列的句子结构来规避检测的攻击。
- **命名实体识别（Named Entity Recognition,NER）：**NER是一种NLP任务，用于识别文本中的命名实体，如人名、组织、地点、医疗代码等。在内容审核中，NER可以用于检测和预防垃圾信息和恶意内容，通过识别已知从事垃圾信息或欺诈活动的公司或个人。此外，NER系统还可以扩展到识别“身份群体”（如性别、种族、性取向、宗教），从而不仅检测有毒语言，还能精确定位受攻击的特定群体。
- **情感分析（Sentiment Analysis）：** 情感分析，也称为意见挖掘，利用NLP和机器学习来识别和提取文本中的主观信息，从而理解其中表达的态度、观点和情绪。它在内容审核中用于识别文本的语气，帮助检测仇恨言论或骚扰等有害内容。
- **主题建模（Topic Modeling）：** 主题建模是一种无监督的NLP方法，通过词语分组来总结文本数据，并揭示文档集中潜在的主题或模式。它在内容审核中可用于识别敏感话题，如仇恨言论和骚扰，以及更一般的主题，从而帮助理解数据中的潜在模式和关系。

多模态人工智能

多模态AI能够同时理解和处理文本、图像、音频和视频等多种输入类型，从而对语境和意图有更丰富的理解。这种技术通过整合不同模态的数据，提高了有害内容检测的准确性，

例如，通过理解某些看似正常的图片与特定文本结合时会变得有害。领先的AI模型，如OpenAI的GPT-Image-1，已具备多模态生成和编辑能力，并集成了企业级安全功能，包括可调谐的审核。

8.3.3 混合式审核（人机协作）

混合式审核结合了AI驱动的自动化和人类监督，是当前内容审核领域的主流和最佳实践。这种“人机协作”（Human-in-the-Loop, HITL）模式在AI系统的整个生命周期中都强调人类的积极参与，包括数据标注、模型训练、验证、部署和持续运营。

HITL模式的运作机制是：AI系统首先识别出其预测置信度较低的案例，然后由人类专家进行审查和验证，再采取实际行动。人类可以根据AI的输出直接干预，调整其行为或修改其输出。这种持续的反馈循环使算法能够不断改进。

HITL模式的优势显著：

- **确保准确性：**人类审核员能够解释语境、多语言文本，并考虑当地市场的文化、区域和社会政治细微差别，从而显著提高审核的准确性和一致性，减少AI可能出现的误报和漏报。
- **减少偏见：** AI系统可能从其训练数据中继承和放大偏见。人类在环能够及早发现和纠正这些偏见。
- **提高效率：**AI能够快速筛选和处理大量数据，将任务传递给人类进行最终的细致审查，从而节省大量时间和成本 。
- **处理复杂和模糊案例：**AI在理解讽刺、幽默或文化引用等细微语境方面仍存在困难。人类审核员能够弥补这一不足，处理需要细致判断的复杂案例。

用户对AI审核的信任也受到人机协作模式的影响。研究表明，让用户参与反馈机制可以增强信任，无论审核源是AI、人类还是两者结合。

8.3.4 内容审核方法总结

下表总结了AI内容审核中使用的关键技术及其主要应用和优势。

表20

技术类别	具体技术	主要应用	优势
AI 模型	基于规则的系统	识别明确违规内容，如特定脏话、垃圾邮件模式	快速、低延迟，适用于简单、明确的违规行为
AI 模型	传统机器学习（如朴素贝叶斯、SVM）	识别明确违规内容，如特定脏话、垃圾邮件模式	快速、低延迟，适用于简单、明确的违规行为
AI 模型	深度学习架构（如 CNN、RNN、LSTM）	复杂文本分类、情感分析、序列数据处理（如聊天记录）	捕捉复杂语义关系，处理长序列数据，克服梯度消失问题
AI 模型	Transformer 模型（如 BERT、GPT、RoBERTa）	语境理解、多语言内容审核、讽刺/俚语检测、虚假信息识别	卓越的语境理解能力，强大的泛化能力，适用于复杂和细致的语言
NLP 技术	文本嵌入	语义搜索、内容相似性检测（如有害信息变体）、推荐系统	将文本转换为向量，捕捉深层语义，实现高效相似性比较
NLP 技术	命名实体识别 (NER)	识别敏感实体（如 PII）、有害内容的攻击目标	精确定位关键信息，增强有害内容检测的粒度

技术类别	具体技术	主要应用	优势
NLP 技术	情感分析	识别文本的情绪倾向与语气，辅助判断仇恨言论/骚扰	理解用户意图与情绪，区分中性与攻击性言论
NLP 技术	主题建模	发现文本数据中的潜在主题与趋势	帮助识别内容热点，辅助平台管理者把握社区讨论方向
其他 AI 技术	多模态 AI	审核结合文本、图像、音频、视频（如仇恨表情包、直播流）	综合理解多种模态信息，提高复杂内容检测的准确性
其他 AI 技术	人机协作 (HITL)	处理 AI 置信度低或复杂/模糊案例，进行最终决策	结合 AI 效率与人类判断力，提高准确性，减少偏见

8.4 AI内容审核面临的挑战

尽管AI在内容审核中展现出巨大潜力，但其应用仍面临多重复杂挑战，这些挑战限制了其有效性并引发了伦理担忧。

8.4.1 语境理解的复杂性

AI系统在理解人类语言的细微差别方面存在固有困难，这常常导致误判。

- **讽刺、反讽与俚语：** AI难以区分字面意义与真实意图相反的表达，如讽刺和反讽。例如，在“这个电影太棒了，演员简直是杀手！”这句话中，AI需要识别“杀手”是比喻而非真实暴力。同样，俚语和新词汇的含义不断演变，使得基于规则的过滤器和传统AI模型难以跟上。
- **多语言与文化敏感性：** AI系统在非英语语言，特别是低资源语言和方言中的表现明显下降。训练数据的盎格鲁中心主义偏见导致AI难以理解当地语境、文化习语和细微差别。例如，某些在一种文化中可接受的内容在另一种文化中可能具有冒犯性。这种语境理解的局限性是AI内容审核的核心挑战，它经常导致误报（无害内容被标记）和漏报（有害内容未被发现），并加剧模型偏见，尤其是在非英语语境中。

8.4.2 模型偏见与公平性

AI模型中的偏见是一个系统性问题，贯穿从数据收集到模型部署的整个生命周期。

- **训练数据偏见：** AI系统通过学习历史数据进行决策，而这些数据可能本身就包含社会和经济不平等、文化、种族或性别偏见。如果这些偏见未得到解决，AI模型可能会强化并放大这些不平等，导致对某些群体的不公平待遇。例如，如果AI系统在偏见数据上训练，它可能会不成比例地压制或错误分类来自弱势群体的言论，甚至将“有毒内容”错误分类为良性。
- **算法偏见：** AI系统的设计本身就可能偏向某些群体，从而强化历史不平等。这可能导致招聘、刑事司法和医疗诊断等领域的歧视性结果。

偏见的系统性存在，从数据收集到部署，都会强化不平等。这不仅损害用户信任，还可能导致法律和声誉风险 。

8.4.3 对抗性攻击

恶意行为者会主动尝试规避AI内容审核系统，这构成了一场持续的“军备竞赛。

- **规避技术：** 攻击者通过注入微小的扰动来精心制作恶意输入，这些扰动对人类几乎不可察觉，但足以欺骗机器学习模型，导致其做出错误的预测或分类。例如，垃圾邮件过滤器可能通过插入细微改变的字符或同义词来规避 。
- **多语言攻击：** AI安全系统的防护措施通常首先以英语设计，这意味着攻击者可以通过切换语言来绕过它们。一些攻击甚至会混合使用多种语言（代码切换），从而混淆AI安全系统。
- **攻击者积极规避系统**，要求防御措施必须持续演进。这需要不断开发更复杂的检测算法和防御策略，如对抗性训练、数据增强和模型集成 。

8.4.4 数据质量与稀缺性

AI模型的有效性直接取决于其训练数据的质量和数量。

- **数据质量：** 糟糕的训练数据会导致AI系统产生不准确或误导性的结果。AI模型无法独立验证训练数据的真实性或其生成输出的有效性，因此开发者有责任确保高质量的数据 。
- **数据稀缺性：** 许多语言缺乏训练有效AI模型所需的大量标注数据集。这种稀缺性阻碍了多语言模型的发展，并导致AI在非英语语境下的审核能力不足。

数据质量差会损害AI的有效性；而**数据稀缺则阻碍了多语言模型的发展**。这强调了投资于强大的数据管理系统、严格的数据治理框架以及高质量、多样化数据集的重要性 。

8.4.5 持续更新与维护

在线内容和有害内容的表现形式不断演变，要求AI系统持续更新和维护。

- **语言演变：** 俚语、新词和表达方式的快速变化要求AI系统不断更新其知识库和模式识别能力。
- **新形式有害内容：** 恶意行为者会不断开发新的规避策略和有害内容形式，如深度伪造（deepfakes）和合成媒体。这要求AI模型具备适应性和持续学习能力。

语言和有害内容不断演变，意味着AI模型需要持续的再训练和适应。这不仅涉及技术挑战，还包括巨大的资源投入，以确保审核系统保持高效和相关性。

8.4.6 误报与漏报

误报（False Positives）和漏报（False Negatives）是AI内容审核中普遍存在且难以避免的问题。

- **误报：** 合法或无害的内容被AI错误地标记为违规并被移除。这可能导致用户沮丧，损害言论自由，并侵蚀用户对平台的信任。
- **漏报：** 有害内容未能被AI检测到，从而传播开来，使用户暴露于风险，并损害平台声誉。

误报与漏报之间存在着固有的权衡，即提高检测敏感度（减少漏报）往往会增加误报，反之亦然。这种权衡对用户信任和平台声誉产生直接影响。解决这一问题需要持续的模型微调、用户反馈机制以及人机协作的混合审核策略。

8.4.7 法律法规与合规性

AI内容审核在中国需要遵守多层次、动态变化的法律法规体系，这对平台提出了复杂的合规挑战。

- **法律多样性与动态调整：**平台需同时满足《网络安全法》《数据安全法》《个人信息保护法》《互联网信息服务管理办法》《互联网信息服务算法推荐管理规定》等要求，这些法规对内容监控、数据使用、算法透明度和用户权利保障提出了不同标准，并随着政策导向持续更新。
- **合规负担与执行压力：**法规普遍要求平台对违法和不良信息承担主体责任，落实“发现即处置”和内容留存报备等义务，若处置不及时或不到位，可能面临罚款、服务下架及信用惩戒。
- **算法透明与责任认定：**AI审核模型在推荐排序和内容筛选中使用时，需确保算法机制透明、合规，且不得滥用算法操纵舆论或侵犯用户权益，同时还要明确错误审核或数据泄漏的责任主体。

这种多维监管和合规压力，要求企业不断调整审核流程和技术架构，在确保合规的同时避免过度干预正常言论，平衡法律责任与用户体验。

8.4.8 AI内容审核面临的挑战与缓解策略

下表总结了AI内容审核中的主要挑战及其对应的缓解策略。

表20

挑战类别	具体挑战	描述	缓解策略
技术局限性	语境理解不足	AI 难以理解讽刺、俚语、文化细微差别和隐含意图。	结合多模态 AI；利用 Transformer 模型进行高级语义理解；加强人机协作以处理复杂语境。
技术局限性	模型偏见	AI 模型从偏见训练数据中学习，导致歧视性或不公平的审核结果。	使用多样化和代表性的训练数据集；实施偏见检测工具和定期审计；开发公平性感知算法。
技术局限性	对抗性攻击	恶意用户通过微小扰动或规避技术欺骗 AI 模型。	实施对抗性训练；采用模型集成；持续更新模型以应对新的规避策略。
数据问题	数据质量与稀缺性	训练数据质量差或数量不足，尤其在低资源语言中更为明显。	投资高质量数据标注；利用合成数据生成；鼓励多语言数据集的创建。
维护成本	持续更新与维护复杂性	语言、俚语和有害内容形式不断演变，要求 AI 模型持续适应。	建立自动化模型再训练管道；实施实时反馈循环；投入研发以提高模型适应性。
维护成本	误报与漏报	AI 错误标记合法内容（误报）或未检测到有害内容（漏报）。	优化模型阈值；引入人机协作进行复审；利用用户反馈微调算法；平衡召回率与精确率。
伦理与法律	言论自由与平台安全冲突	在保护言论自由与维护平台安全之间找到平衡点。	制定清晰透明的社区准则；提供用户申诉机制；加强人机协作以进行细致判断。
伦理与法律	透明度与问责制不足	AI 决策过程不透明，用户难以理解内容被标记或移除的原因。	实施可解释 AI (XAI) 技术；提供详细审核决策解释；发布透明度报告。
伦理与法律	用户隐私与数据保护	内容分析涉及大量用户数据，引发隐私泄露风险。	采用数据最小化原则；使用数据匿名化技术；探索联邦学习等隐私保护技术。

8.5 游戏UGC安全审核的新尝试

8.5.1 业务背景及痛点剖析

作为国内游戏行业头部开发商，M游戏公司始终坚持探索游戏新玩法，其中也包括了在已上线的游戏中结合游戏特色进行玩法创新。自2024年起，游戏项目组开始探索在游戏中加入UGC（User Generated Content，用户生产内容）的元素和交互模式——用户可在自有的家园系统空间中，通过有限的素材创作3D作品，并支持作品的一键复制和社区分享。这个玩法极大地提升了用户在游戏中的自由度，有利于保持游戏活力，但较高的自由度也带来了各种挑战，内容安全审核就是其中一道绕不开的坎。

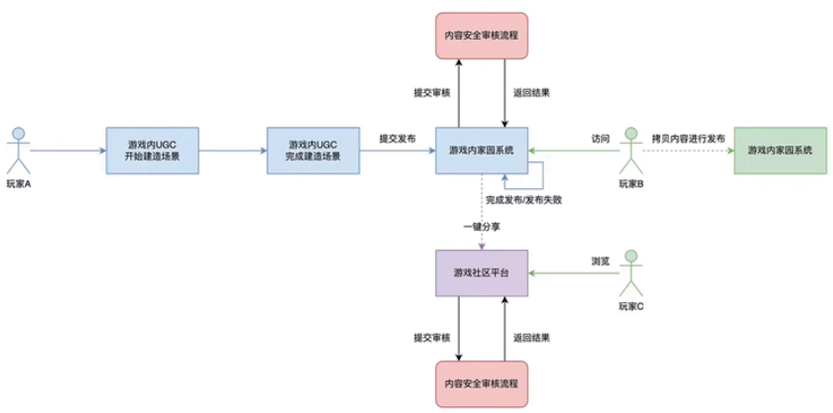

图67

8.5.2 UGC内容安全审核场景痛点剖析

海量UGC带来的审核工作量剧增

作为一款高DAU网络游戏，用户对新玩法都有一定的热度，随之而来便是UGC数量激增，一天内上报的审核请求数少则数万多则数十万。除此之外，游戏玩法中各种设定也鼓励用户不断更新作品以维持玩法热度，单一用户每次更新发布后触发作品重新审核，使得该模式开放后审核请求数居高不下。项目组在UGC玩法上线前，曾对所需的人力成本进行测算，预估需要维护50人的客服团队才能满足UGC正式上线后的审核需求——这对UGC玩法上线是一个巨大的成本挑战。

玩家热度波动导致的审核需求波动

受游戏版本更迭、游戏活动、工作日与节假日游玩时间差异等多因素影响，游戏DAU必然会出现波动。而UGC并非游戏的核心内容，大部分用户都是在登录游戏体验主要内容的过程中“抽空”进入家园系统进行UGC创作，因此UGC审核需求量也会在每天甚至每个小时出现波动。如全部采用人工审核，如何在兼顾整体审核效率的情况下合理分配审核人员资源将会是一个难题。

人工审核的主观性和准确率风险

即使所有审核人员都经过统一规范培训、审核流程中有明确的审核标准，但人工审核过程中依旧无法完全规避个人主观性带来的审核偏差或准确率过低风险。

8.5.3 技术方案设计与落地

UGC内容安全审核工程链路设计

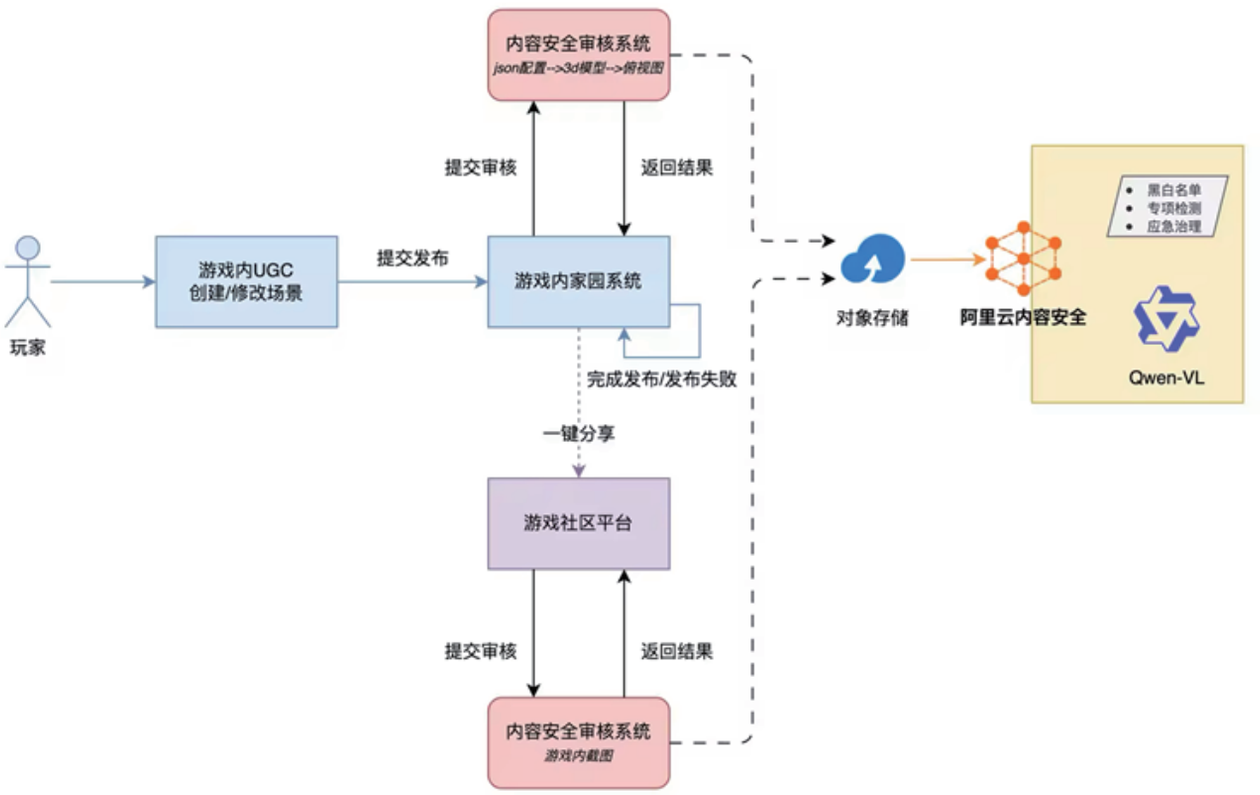

图68

游戏业务视角

- 玩家在游戏内提交发布后，游戏服务器自动将其家园系统JSON配置文件转为3D建模，然后进一步生成俯视图文件并存入OSS对象存储；此时家园系统状态为“审核中”，玩家也可进入家园中继续编辑、保存草稿、发布新作品。
- 如果玩家将家园系统截图分享至社区平台，社区平台服务器则直接将图片内容存入OSS对象存储，此时帖子状态为“审核中”。
- 游戏内容安全审核系统根据玩家动作信息流，调用阿里云内容安全SaaS服务API，对存入OSS的图片发起内容审核。
- 阿里云内容安全SaaS进行同步/异步审核。
- 阿里云内容安全SaaS审核完毕后，返回审核结果，根据不同的审核结果做出不同的业务动作：
  - 如审核结果为“通过”，游戏服务器将玩家家园系统置为“已更新”，并允许其他玩家进行访问
  - 如审核结果为“不通过”，游戏服务器将玩家家园系统置为“未更新”，并通知玩家发布结果
  - 如审核结果为特定类型（如“疑似遮挡”），游戏内容安全审核系统将进行其他动作（如：重新生成俯视图再次提交审核、转入人工审核工作流程...）

内容安全&Qwen-VL视角

- 用户程序调用内容安全SaaS服务API，对存入OSS的图片发起内容审核
- 内容安全SaaS服务根据预设规则、提示词、其他参数，调用Qwen-VL模型能力对图片进行内容审核，并获得答复
- 内容安全SaaS服务基于Qwen-VL答复和其他服务配置（黑白名单、检测范围...），产出最终审核结果
- 用户程序调用内容安全SaaS服务API，查询内容审核结果

内容安全+Qwen-VL的产品服务形态

- 作为一款成熟可靠的SaaS服务，阿里云内容安全在M游戏公司已有的多个业务场景中展现出良好的稳定性、检测效果和用户体验。在这次项目中，M游戏公司业务安全团队使用了阿里云内容安全标准API进行服务调用与结果查询，**可以极低的业务改造成本快速启动POC，极大加速了项目进程。**
- Qwen-VL通过优秀的图像理解、推理能力进一步加强了内容安全SaaS服务已有的图片审核能力，并具备通过SFT或其他手段优化特定专业领域图片审核表现的可能性，**为丰富多变的UGC安全审核场景提供更高能力上限。**
- 内容安全不仅降低了客户业务代码复杂度，也降低了客户对Qwen-VL模型调优的成本和技术过程（如：提示词工程优化、SFT微调）。**客户只需关注业务效果的测试、效果评价与效果反馈。**

8.5.4 项目成功落地后带来的业务收益

审核效果远高于人工审核的同时，重保敏感违规审核效果

结合了Qwen-VL模型能力的内容安全服务，在该场景下的审核准召率远高于人工审核，在为期一个月的对比测试中，人工审核对负样本召回率为52%，而模型审核负样本召回率达到了95%。在保障审核效果的同时，在模型能力加持下能重点保障涉政类型的负样本召回率。

模型审核将审核召回量控制在较低水平，有效降低了业务综合成本

在内容安全+Qwen-VL的方案可行性与质量均通过测试后，游戏内容安全审核团队重新评估了所需投入的人力成本，仅需10人即可满足UGC玩法上线后的审核工作量，包含：a）对模型召回的负样本进行复审；b）对模型未召回的样本抽检，并进行内容安全相关工作的数据挖掘与积累。相比之前预估的方案（50人客服团队），模型审核将人力成本降低了80%。

伴随着游戏新版本上线，游戏中的自由建造玩法向全区服玩家开放，大量玩家开始在游戏中创建自己的家园系统，社交媒体中也陆续出现了相关话题交流和分享，部分视频播放量达到10W+甚至100W+，玩法热度可见一斑。

8.6 未来趋势与展望

内容审核在游戏场景演着不可或缺的角色，是维护游戏平台安全、促进健康交流和确保商业可持续性的基石。随着用户生成内容以惊人的速度增长，传统的人工审核模式已无法满足需求，这促使人工智能成为审核流程中不可或缺的组成部分。AI系统以其无与伦比的速度和可扩展性，能够高效地处理海量数据，识别并过滤各类有害内容，从明确的非法信息到复杂的仇恨言论和虚假信息。

AI在理解人类语言的细微差别、语境、讽刺和文化敏感性方面仍面临显著障碍，这常常导致误报和漏报。此外，AI模型中固有的偏见，源于训练数据的偏差和算法设计，可能导致歧视性结果，并损害公平性。对抗性攻击的持续存在，以及语言和有害内容形式的不断演变，都要求AI系统必须持续更新和维护，形成一场永无止境的“军备竞赛”。

鉴于这些挑战，纯粹的自动化审核并非终极解决方案。**人机协作的混合式审核模式**是实现高效、公平和负责任内容审核的关键。AI负责处理高流量、明确的违规内容，从而显著提高效率和一致性；而人类审核员则专注于处理AI置信度较低的复杂、模糊和语境敏感的案例，提供AI所缺乏的判断力、同理心和文化理解。这种互补关系不仅提高了审核的准确性，也减轻了人类审核员的心理负担。

展望未来，AI内容审核将继续向更高级的语义理解、多模态内容整合和更强的隐私保护方向发展。AI生成内容的爆炸式增长将带来新的审核复杂性，促使AI技术不断创新以应对其自身创造的挑战。同时，全球范围内不断演变的法律法规将进一步塑造内容审核的实践，推动平台在技术、伦理和法律层面进行全面的调整。

8.6.1 高级语义理解

未来的AI内容审核将超越简单的关键词匹配，实现对人类语言的**高级语义理解**。

- **深度语境感知：** 随着Transformer模型和大型语言模型（LLMs）的进步，AI将能够更深入地理解文本的语境、意图和细微差别 。这意味着AI将能更好地识别讽刺、隐喻、俚语和文化习语，从而减少误报和漏报 。
- **情感推理：** 结合情感分析和情绪识别技术，AI将能够理解内容的情感影响，从而更精准地检测微妙的骚扰或隐藏的威胁 。

这种能力将使AI能够从关键词识别迈向对意图和细微差别的真正理解，从而实现更精准、更具情境感知的审核。

8.6.2 多模态人工智能的扩展

多模态AI的融合是内容审核的另一个重要趋势，它将整合文本、图像、音频和视频等多种数据类型，以实现更全面的语境理解。

- **综合内容分析**： 未来的AI系统将能够同时处理和分析不同模态的信息，例如，识别结合了文字和图像的有害表情包 。这将显著提高对复杂和隐蔽有害内容的检测能力 。
- **实时处理：** 随着计算效率的提升，多模态AI将能实现对直播流和实时聊天的即时审核，

从而有效阻止有害内容在第一时间传播 。

整合文本、图像、音频和视频将实现对内容的整体语境理解，从而提升审核的深度和广度。

8.6.3 联邦学习与隐私保护

随着数据隐私法规日益严格，联邦学习（Federated Learning, FL）将成为内容审核中的重要技术。

- **分布式模型训练：** FL允许AI模型在不集中收集原始训练数据的情况下，跨多个设备或服务器进行训练 。参与者在本地数据上训练模型，然后只将模型更新发送到中央服务器 。
- **增强隐私：** 这种方法显著降低了数据保护风险，因为敏感的用户数据无需离开其来源设备 。它有助于平台遵守GDPR等严格的隐私法规 。

联邦学习通过在**不集中化敏感数据的情况下训练模型**，有效解决了数据隐私问题，同时仍能提升模型的性能和泛化能力。

8.6.4 AI生成内容的审核

随着生成式AI（Generative AI）技术的快速发展，AI生成的内容（如深度伪造、合成媒体）对内容审核提出了新的、复杂的挑战。

- **“军备竞赛”：** 生成式AI能够以极高的质量和数量快速生成欺骗性内容，包括图像性虐待内容或误导性信息 。这导致了一场AI生成内容与AI检测内容之间的“军备竞赛” 。
- **检测复杂性：** 检测AI生成内容需要AI审核系统具备更高的复杂性和适应性，能够识别出人类难以察觉的细微模式和伪造痕迹 。

AI生成内容必然需要AI来检测，这导致了一场持续的“军备竞赛”，推动着AI审核技术的不断创新和迭代。

8.6.5 不断演变的监管框架

全球范围内对AI和内容审核的立法努力将深刻塑造未来的审核实践。

- **全球协同：** 欧盟的《数字服务法案》和《人工智能法案》等法规正在推动平台对内容审核实践承担更多责任，并要求更高的透明度和问责制 。
- **平衡与挑战：** 监管机构正努力在保护言论自由、遏制有害内容和促进技术创新之间找到平衡 。

全球立法努力将塑造未来的内容审核实践，促使平台在技术、伦理和法律层面进行全面调整，以适应日益复杂的数字治理环境。

第九章：游戏行业的昨天、今天与明天

如果将游戏行业比作一艘航船，那么过去是它破浪启航的序章，今天是它乘风破浪的征程，而明天，则是驶向未知蓝海的壮阔图景。从像素点构成的《Pong》到如今万人同服的虚拟世界，从单机本地运行到全球云原生部署，游戏已不仅仅是娱乐，更成为技术革新、文化表达与社会连接的重要载体。最后回望游戏行业的技术演进之路，梳理当前云上实践的核心成果，并展望未来十年可能重塑行业的关键技术趋势。

9.1 昨天：从本地化到互联网化——游戏的“基建时代”

20世纪末至21世纪初，是游戏行业的“奠基期”。这一阶段的核心特征是：

- 架构本地化：游戏服务器部署在物理机房，扩展性差，灾备能力弱，运维成本高。
- 网络依赖有限：早期单机游戏主导市场，联网游戏多为局域网对战，延迟与带宽问题尚未凸显。
- 安全防护初级：DDoS攻击尚不普遍，内容审核多靠人工，数据丢失风险高。
- 美术依赖人力：角色、场景、动画全部由美术团队手工制作，周期长、成本高。

那时的游戏开发更像“手工作坊”，技术服务于玩法实现，而稳定性、可扩展性、全球化运营并非优先考量。然而，正是这些“痛点”催生了后续的技术变革。

9.2 今天：云原生驱动——游戏的“智能运营时代”

进入2020年代，云计算、大数据、AI等技术的成熟，推动游戏行业迈入“云原生时代”。本书前八章所探讨的每一个主题，正是当下行业主流实践的缩影：

- 架构弹性化：微服务、容器化（Kubernetes）、Serverless 架构让游戏系统具备分钟级扩缩容能力，轻松应对开服高峰。
- 网络全球化：通过全球加速、边缘节点、智能DNS调度，实现跨洲低延迟对战，真正支持“全球同服”。
- 安全体系化：从被动防御转向主动监控，结合AI行为分析、流量指纹识别，实现毫秒级攻击拦截。
- 数据驱动化：湖仓一体架构、实时计算引擎（Flink）让玩家行为分析从“事后复盘”变为“实时干预”。
- 内容智能化：AIGC辅助美术生成、大模型参与剧情设计、智能NPC对话系统逐步落地。
- 审核自动化：第八章所述的多模态内容审核体系，结合大模型语义理解，显著提升UGC生态的安全与效率。

今天的成功游戏，已不仅是“好玩”，更是“好技术 + 好运营 + 好生态”的综合体现。云平台不再只是资源提供者，而是成为游戏全生命周期的“数字底座”。

9.3 明天：融合与颠覆——游戏的“虚实共生时代”

展望未来，游戏将不再局限于“屏幕内的娱乐”，而是向更广阔的“数字生活空间”演进。以下几个趋势，或将重新定义游戏的边界：

9.3.1 全面云化与无端化（Cloud Gaming 2.0）

随着5G/6G、边缘计算和编解码技术的进步，云游戏将摆脱“延迟高、成本大”的桎梏。未来玩家无需下载客户端，打开浏览器或AR眼镜即可进入3A级游戏世界，真正实现“即点即玩”。

9.3.2 AI深度融入游戏核心逻辑

AI Game Master：模型将扮演“动态剧情导演”，根据玩家行为实时生成任务、NPC对话甚至世界观演变。

智能外挂对抗：AI不仅能检测作弊，还能模拟外挂行为，用于自动化攻防测试。

个性化体验：AI根据玩家性格、习惯、情绪（通过生物识别）动态调整难度与叙事节奏。

9.3.3 UGC 3.0与创作者经济爆发

在AI工具加持下，普通玩家也能轻松创作高质量地图、剧情、角色。游戏平台将演变为“创作生态”，通过NFT、链上确权、智能合约实现创作者收益分成，形成真正的“玩家共建世界”。

9.3.4 虚实融合：游戏作为元宇宙入口

游戏将成为元宇宙的核心场景之一。虚拟身份、数字资产、社交关系将在多个游戏与应用间互通。VR/AR设备的普及，将让“沉浸式交互”成为常态，游戏与教育、办公、社交的界限逐渐模糊。

9.3.5 审核与治理的“自治化”

面对海量UGC与AI生成内容，传统审核模式将难以为继。未来可能出现“去中心化审核网络”或“AI自治社区”，通过共识机制与智能合约实现内容治理，平衡自由与安全。

9.4 技术为舟，创意为帆

回顾游戏行业的昨天，我们看到了技术从“支撑”到“驱动”的转变；审视今天，云原生、AI、大数据已深度融入游戏开发与运营的每一个环节；展望明天，游戏将不再只是一个产品，而是一个持续进化的“生命体”——它连接人与人，融合现实与虚拟，承载文化与想象。本书所探讨的架构、网络、安全、数据库、大数据、美术、内容审核等，正是这艘航船的“龙骨”与“风帆”。无论技术如何演进，其终极目标始终未变：为玩家创造更自由、更安全、更精彩的世界。未来已来，唯变不破。愿每一位游戏人，都能在这场数字文明的浪潮中，乘风破浪，驶向属于我们的星辰大海。