2021年
全球Serverless服务洞察

头豹研究院
弗若斯特沙利文咨询（中国）

报告说明

沙利文谨此发布中国云原生系列报告之《2021年全球Serverless服务洞察》年度报告。本市场报告以Serverless产品为核心研究对象，研究周期覆盖2020年全年。本研究项目将对Serverless市场动向、前沿技术、商业模式、竞争态势等信息进行重点梳理，并从应用场景拓展、生态价值创造等维度出发对市场发展前景做出推测或预判。

本研究项目旨在梳理Serverless服务体系，洞悉用户特点、市场存量空间及增量空间，并结合市场发展前景判断Serverless行业变革。

本报告所有图、表、文字中的数据均源自弗若斯特沙利文咨询（中国）及头豹研究院调查，数据均采用四舍五入，小数计一位。

报告提供的任何内容（包括但不限于数据、文字、图表、图像等）均系弗若斯特沙利文及头豹研究院独有的高度机密性文件（在报告中另行标明出处者除外）。未经弗若斯特沙利文及头豹研究院事先书面许可，任何人不得以任何方式擅自复制、再造、传播、出版、引用、改编、汇编本报告内容，若有违反上述约定的行为发生，弗若斯特沙利文及头豹研究院保留采取法律措施、追究相关人员责任的权利。弗若斯特沙利文及头豹研究院开展的所有商业活动均使用"弗若斯特沙利文"、"沙利文"、"头豹研究院"或"头豹"的商号、商标，弗若斯特沙利文及头豹研究院无任何前述名称之外的其他分支机构，也未授权或聘用其他任何第三方代表头豹研究院开展商业活动。

- **Serverless（无服务器）是一种具有隐藏服务器、事件驱动、持续伸缩、按照调用付费等特征的云原生开发模型**

Serverless面向开发者时屏蔽了服务器的复杂度，开发者无需针对服务器进行繁琐开发和运维操作；Serverless FaaS中的函数Function运行在独立的容器里，基于事件驱动；Serverless提供一种按照调用付费的计费模型，并且在资源空闲时不收费；此外，Serverless提供极强的弹性伸缩能力，让资源适配业务需求。

- **Serverless营收规模快速扩增，2020年全球市场营收达446.1亿元；Serverless现阶段主要面向IT、金融及公共部门等领域**

2016-2020年，Serverless服务在全球的热度持续增长，市场营收规模年复合增长率达到28.0%。2020年全球Serverless服务市场规模达到446.1亿元。中国Serverless市场的扩张与全球市场基本一致。2020年中国Serverless服务市场规模达到63.7亿元，在全球市场中占比约为14.3%，2025年中国Serverless营收规模预计将达218.2亿元。

- **Serverless发展趋势预测**

Serverless将渗透至大部分服务器场景中；Serverless与容器两大云原生技术将深度融合；Serverless会针对全环节进行优化。

目录

- Serverless技术特征 ---------- 05
- Serverless市场空间 ---------- 08
- Serverless发展趋势 ---------- 12
- 名词解释 ---------- 15
- 方法论 ---------- 16
- 法律声明 ---------- 17

东方财富 leadleo.com

图表目录

- Serverless架构 ---------- 05
- 传统架构与无服务器架构的区别 ---------- 06
- Serverless关键特性 ---------- 07
- Serverless优劣势对比 ---------- 07
- 全球Serverless服务市场规模（按营收计），2016-2025年预测 ---------- 08
- 中国Serverless服务市场规模（按行业拆分，按营收计），2016-2025年预测 ---------- 09
- 全球Serverless市场供应商份额，2020年 ---------- 11
- 中国Serverless市场供应商份额，2020年 ---------- 11
- 中国Serverless市场下游行业营收占比，2020年 ---------- 11
- Serverless中特定场景与通用场景的对比 ---------- 12
- Serverless容器运维边界 ---------- 13
- Serverless与云原生生态 ---------- 14

东方财富 leadleo.com

技术特征

“Serverless是一种具有隐藏服务器、事件驱动、持续伸缩、按照调用付费等特征的云原生开发模型”

什么是无服务器Serverless?

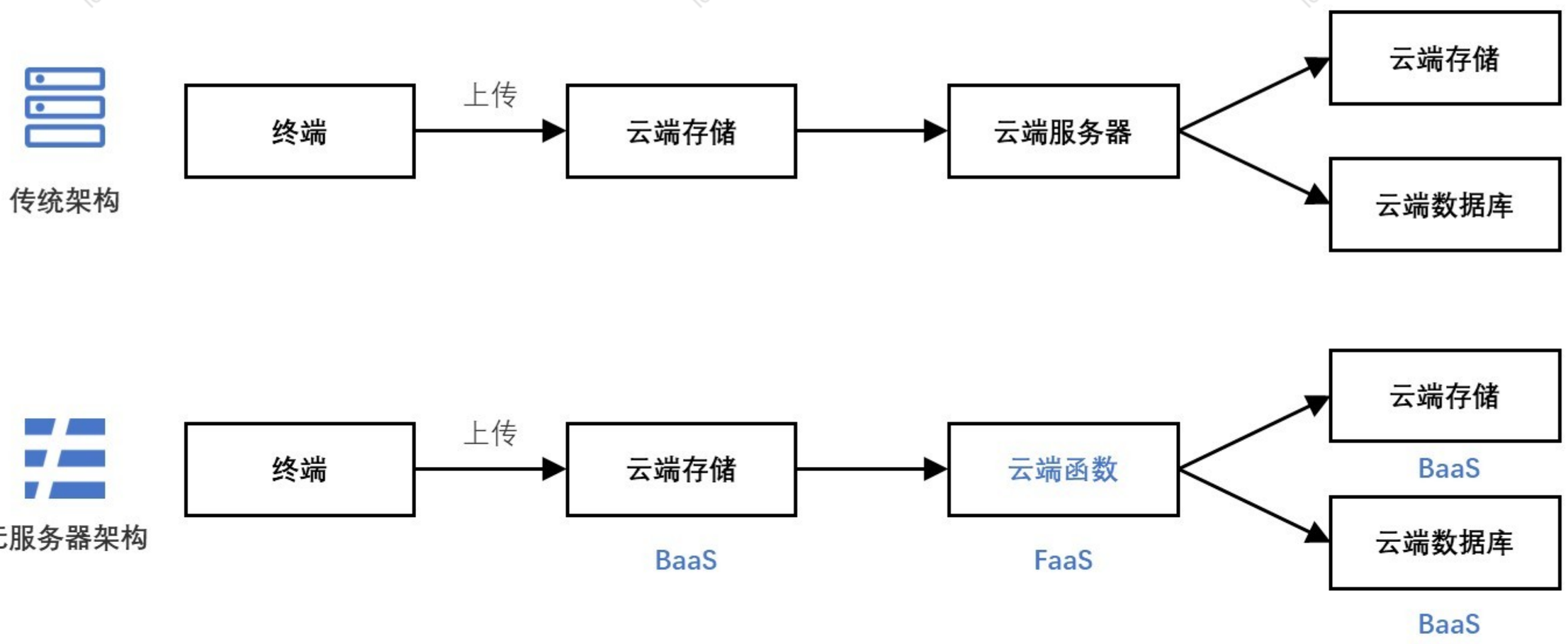
- 无服务器（Serverless）是一种云原生开发模型，可使开发者在无需管理服务器的情况下专注构建和运行应用。
- 无服务器方案将服务器从应用开发中抽离，由云供应商负责置备、维护和扩展服务器基础架构，而开发者则可仅将代码打包到容器中进行部署。在开发者的代码部署后，无服务器应用即可响应需求，并根据需要自动扩容。公共云供应商的无服务器产品通过事件驱动执行模型以按需计量。因此，当无服务器功能闲置时不会产生费用。

Serverless架构

- 无服务器架构主要包括函数即服务（Function as a Service，FaaS）与后端即服务（Backend as a Service，BaaS）两个方面：
  - FaaS包含服务器端业务逻辑的无状态Function，其运行在独立的容器中。这些Function基于事件驱动，并由第三方云供应商托管。
  - BaaS使用第三方服务，大多是富客户端应用程序，如移动App等。客户端负责处理大部分的业务逻辑，其他部分如认证、数据库、用户管理等则依赖外部服务。
- 在传统架构中，业务部署流程通常包括开发业务代码、购买服务器资源、按照相关软件配置相关参数、部署应用与运维管理等环节。与传统架构相对比，无服务器的业务流程简化为开发业务代码与部署应用两个方面。传统架构需要非常长的处理和搭建流程，而在无服务器架构中开发者仅需负责FaaS的代码处理逻辑即可，云供应商会提供BaaS服务的配合，以实现完整的Serverless架构。

来源：InfoQ，弗若斯特沙利文

传统架构与无服务器架构的区别



Serverless意义何在？

- 从最初大型机时代应用定制化的硬件和软件，到PC时代应用的CPU、内存、硬盘、主板、USB 设备等标准化的硬件及软硬件解耦后操作系统、库等可复用组件，再到云时代硬件软件化及软件服务化的趋势，**云服务器是计算机技术"抽象、解耦、集成"化发展的必然趋势。**
- 随着计算功能架构的演进，计算功能愈发强大，而开发者需关注的内容也逐渐聚焦于最核心功能，Serverless让开发者仅需关注应用端最核心的功能——代码：
  - 在计算资源从最初的本地物理机进化到本地虚拟机的过程中，算力资源使用的弹性和灵活度得以提升；
  - 而在从本地虚拟机向云端迁移后，软硬件解耦合的程度加深。开发者无需管理数据中心、网络等资源，且可在不投入本地资源成本的同时拥有近乎无限的算力资源；
  - 云端容器化架构让开发者无需关注操作系统，仅需关注代码与代码运行环境；
  - Serverless的出现在容器化架构的基础上进一步简化开发者的部署流程，让开发者的关注点聚焦于代码端。
- Serverless可以屏蔽基础设施的复杂度，基于云服务快速构建应用，是全托管的计算服务，客户编写代码构建应用，无需管理和运维服务器等底层基础设施。
- 基于Serverless的架构原理，Serverless可广泛应用于小程序/Web/Mobile/API等后端服务、大规模批处理任务处理、基于事件驱动架构的在线应用和离线数据处理以及运维自动化等场景。

来源：阿里云，弗若斯特沙利文

Serverless关键特性：隐藏服务器、事件驱动、持续伸缩、按照调用付费

- Serverless具有如下的关键特征：
  - Serverless隐藏了服务器的概念。Serverless面向开发者时屏蔽了服务器的复杂度，开发者无需针对服务器进行繁琐开发和运维操作；
  - Serverless FaaS中的函数Function运行在独立的容器里，基于事件驱动；
  - Serverless提供一种按照调用付费的计费模型，并且在资源空闲时不收费；
  - Serverless提供极强的弹性伸缩能力，让资源适配业务需求。

Serverless关键特性

Serverless优劣势对比

优势	劣势
服务器管理成本低	第一次调用函数会存在延迟
运营成本低	企业对服务器缺乏控制
业务弹性强	调优仅可通过代码一种途径
事件驱动具备安全性	具有执行时间限制
成本由请求或事件数决定	开发环境与生产环境不一致
部署流程极为简化	测试环节复杂程度高
发布周期缩短	调试环节复杂程度高
监控开箱即用	成本不确定性

来源：InfoQ，弗若斯特沙利文

市场空间

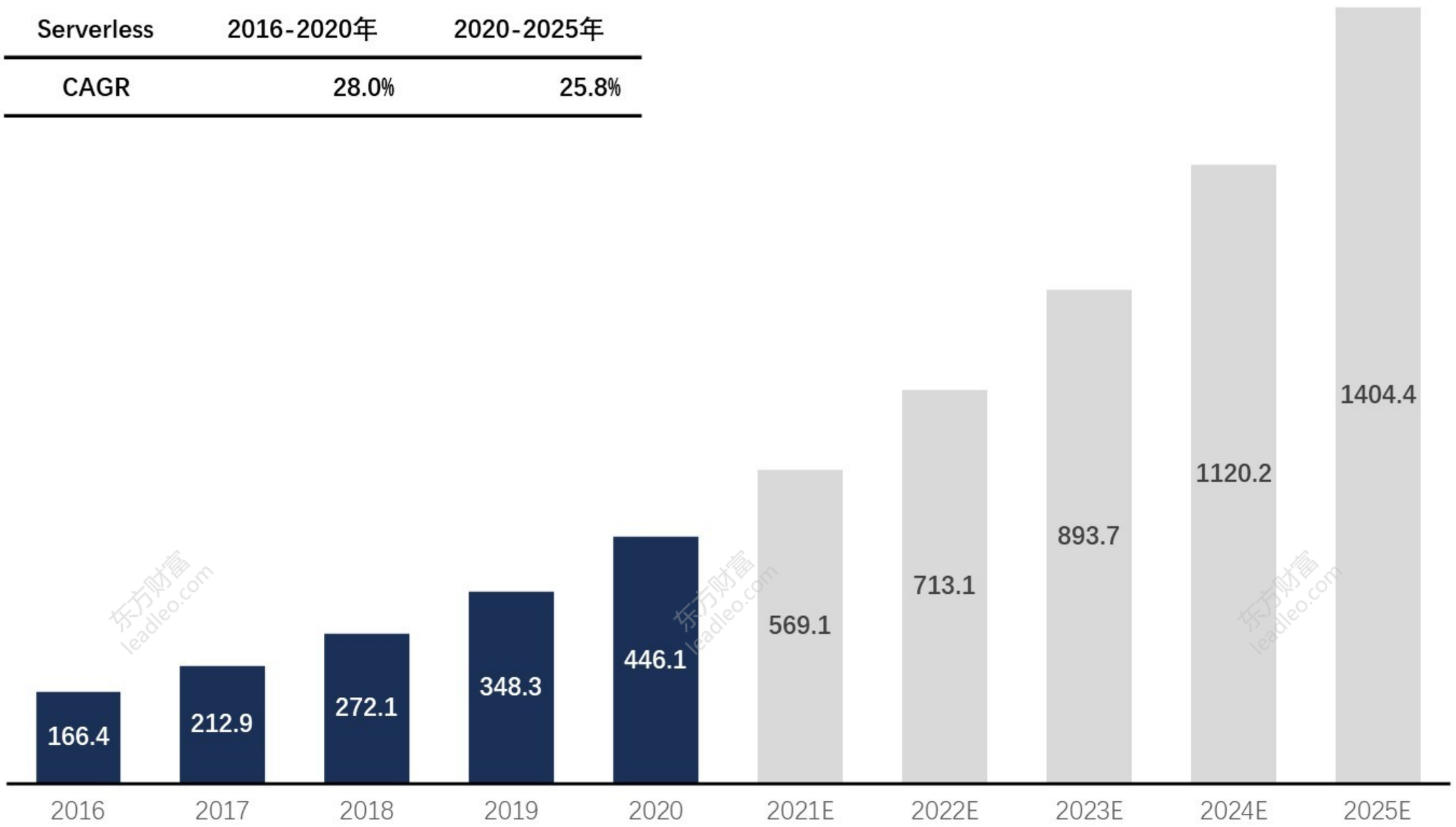
“Serverless营收规模快速扩增，2020年全球市场营收达446.1亿元；Serverless现阶段主要面向IT、金融及公共部门等领域”

- 2020年全球Serverless服务市场规模达到446.1亿元。 2016-2020年，Serverless服务在全球的热度持续增长，市场营收规模年复合增长率达到28.0%。
- Serverless服务可广泛应用于一方云服务、三方SaaS服务以及用户自建的系统中，Serverless的渗透率仍存在较高的增长空间。此外，Serverless将与云原生的其他技术更紧密融合，提升自身服务的性能，扩大营收服务潜力。随着Serverless架构的成熟及渗透率的增长，未来Serverless总体营收增速仍将保持较高水准，预计2025年全球Serverless营收规模将达1404.4亿元。
- 中国Serverless市场的扩张与全球市场基本一致。2020年中国Serverless服务市场规模达到63.7亿元，在全球市场中占比约为14.3%。2025年中国Serverless营收规模预计将达218.2亿元，营收的年复合增长率略高于全球平均水平，在全球市场中占比预计将小幅提升（15.5%）。

全球Serverless服务市场规模（按营收计），2016-2025年预测

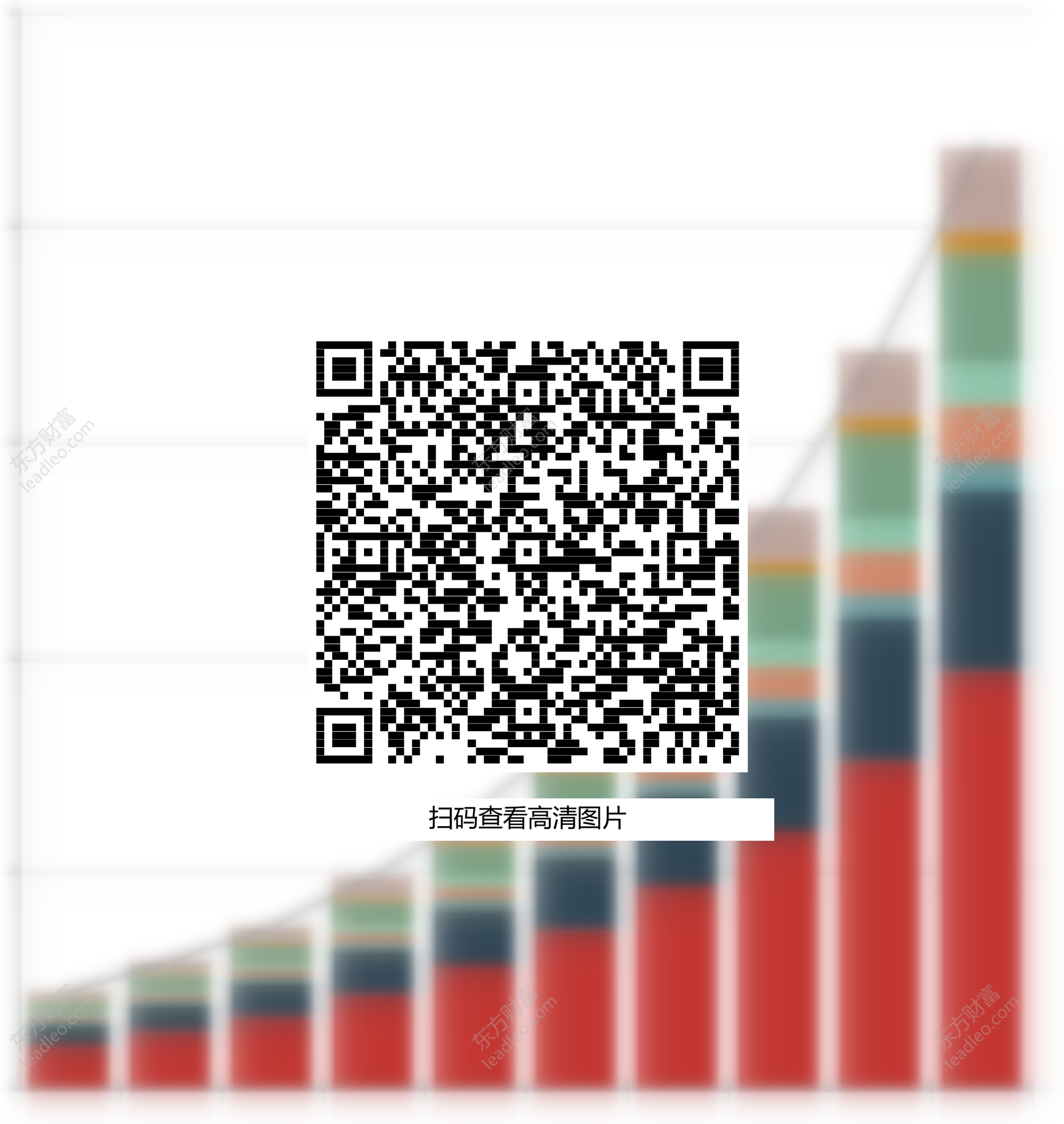
单位：亿元人民币

Serverless	2016-2020年	2020-2025年
CAGR	28.0%	25.8%



来源：弗若斯特沙利文

中国Serverless服务市场规模（按行业拆分，按营收计），2016-2025年预测



https://www.leadleo.com/sizepro/details?id=611342c8c4ba995f57a772ac&core=6126cb800dd177b8f5538dd3

来源：弗若斯特沙利文

- 在全球Serverless市场中，最大的两家供应商分别是Amazon AWS与Microsoft Azure。AWS与Azure在Serverless中布局最早，且背靠云计算渗透率更高、规模更大的北美市场，因此AWS与Azure在市场份额保持领先。全球Serverless头部厂商的基础架构与服务特征基本类似：

  - AWS的无服务器应用程序的托管存储库AWS Serverless Application Repository可以发现、部署和发布无服务器应用程序。在AWS的Serverless服务中，每个应用程序均使用AWS无服务器应用程序模型 (SAM) 模板打包，开发者使用无服务器应用程序存储库不会收取额外费用，开发者只需为其部署的应用程序中使用的AWS资源付费。

  - Azure Serverless可根据工作负载量自动灵活地缩放，将重点放在增值而不是基础结构管理上。基于触发器和绑定的集成编程模型，帮助开发者响应事件并无缝连接到其他服务，并为开发者提供端到端的开发体验，提供各种编程语言和托管选项，从构建、调试到使用集成工具和内置DevOps功能进行部署和监控。

- 阿里云是最早提供Serverless计算服务的云计算厂商之一，市场份额排名中国厂商之首，位列全球第三：

  - 阿里云Serverless服务面向Web后端服务、大数据分析处理、IoT后端服务、集成多产品、实时数据流处理等场景，提供函数计算、对象存储、API网关、表格存储、日志服务、批量计算等服务模式。阿里云函数计算FC可支撑超过百万函数、月调用数千亿次，可实现50ms冷启动、3ms热恢复、百毫秒弹性，可稳定支撑阿里巴巴“双十一”千万级QPS峰值。2020年，中国信息通信研究院发布《云原生用户调查报告》中披露阿里云Serverless产品在中国Serverless用户规模的占比达到66%，超过其他云厂商总和。

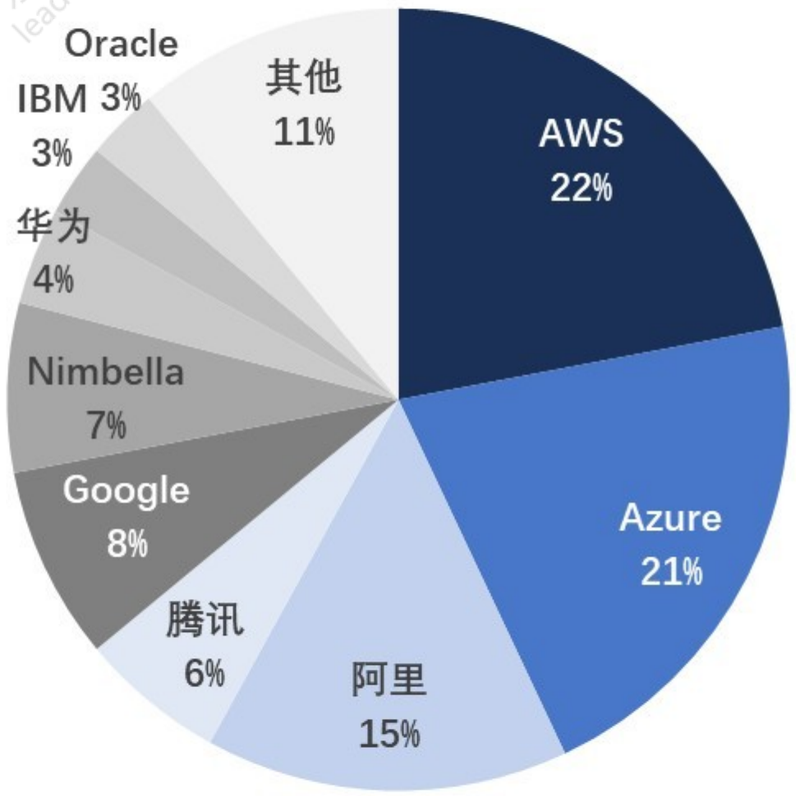
- 中国Serverless市场中主要供应商还包括腾讯与华为，腾讯与华为的Serverless架构能力在行业中均处于领先地位。

  - 腾讯云与Serverless.com战略合作，基于无服务器应用框架Serverless Framework，为用户提供Serverless应用开发管理平台。开发者无需关心底层资源，即可快速部署完整可用的Serverless应用架构，具有资源编排、自动伸缩、事件驱动等能力，覆盖编码-调试-测试-部署等全生命周期，帮助开发者通过联动云资源，迅速构建Serverless应用。

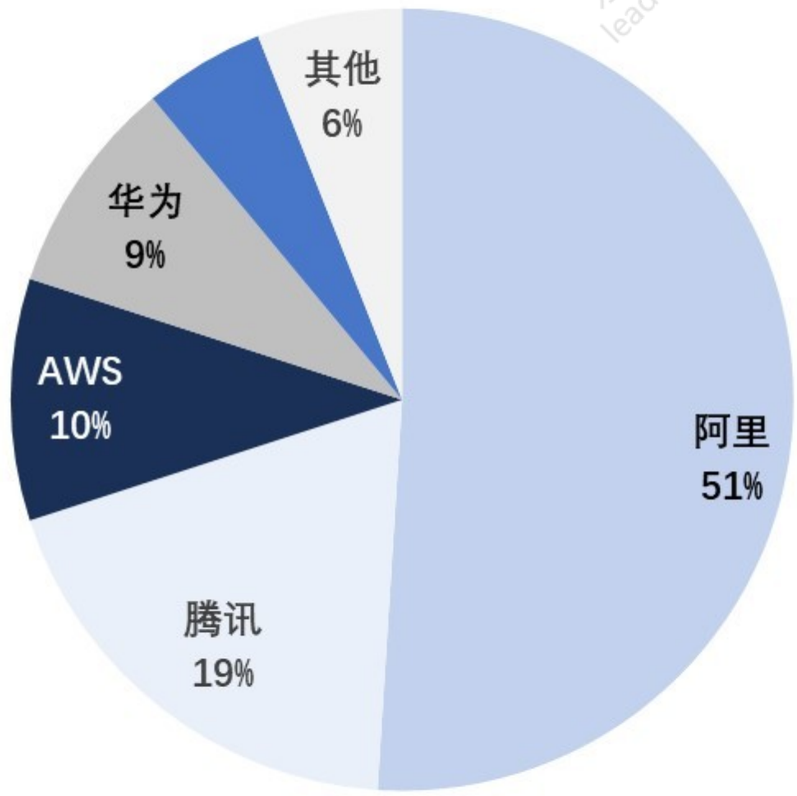
  - 华为云Serverless函数工作流(FunctionGraph)通过了基础能力要求、平台可观测能力、服务性能、服务安全和服务计量准确性等五大类、20+项测试，以稳定、可靠、高效的服务能力荣获可信云函数即服务能力认证。华为云推出带编排能力的函数计算服务的Serverless函数工作流FunctionGraph，为开发者提供界面化管理、一站式的函数开发上线功能，并可通过事件触发集成多种云服务，满足不同场景需求。

来源：中国信通院，阿里云，弗若斯特沙利文

全球Serverless市场供应商份额，2020年



中国Serverless市场供应商份额，2020年



- 由于Serverless可有效减少运营成本、按序自动伸缩且用户不用再关注维护期的运维问题，企业对于Serverless的接纳速度较快，其中大型企业或组织对于Serverless的接纳程度更高。
- Serverless的主要应用领域包括IT、金融以及公共部门等，此外，Serverless在制造、能源、医疗、零售等行业中的收入规模也较可观。未来，金融企业端Serverless的营收具有更高的潜力，其原因是金融领域中出现了许多金融科技初创企业，这些FinTech公司对于云原生及Serverless服务的接纳程度更高。
- Serverless在各场景中落地仍存在一些挑战，例如员工的教育与培训成本高、Serverless领域缺少标准、用户容易被云服务商锁定（迁移难度大）以及代码安全问题。

中国Serverless市场下游行业营收占比，2020年

扫码查看高清图片

https://www.leadleo.com/sizepro/details?id=61134ad3c4ba995f57a77332&core=6126cb870dd17794ba538e0b

来源：弗若斯特沙利文

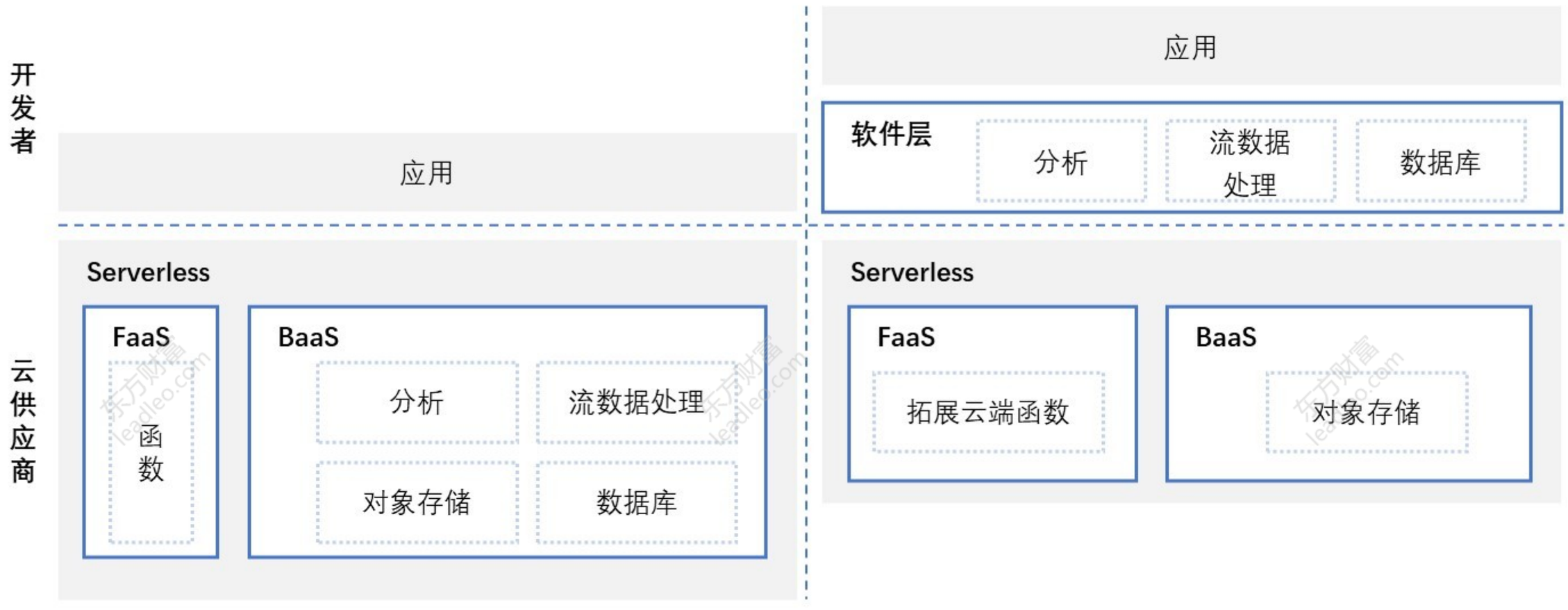
发展趋势

“Serverless将渗透至大部分服务器场景中；Serverless与容器两大云原生技术将深度融合；Serverless会针对全环节进行优化”

发展趋势一：特定场景 + 通用场景 = Serverless everywhere

- Serverless的服务主要分成特定场景与通用场景两部分。
  - 在特定场景下，开发者可在弹性伸缩的平台中实现数据库输入、实时数据队列、机器学习等特定操作。在特定场景中用户业务代码需遵循平台限制，例如运行环境、运行时长、无GPU加速等。随着平台的技术优化，特定场景下的诸多限制会被逐步放宽。
  - 在通用场景下，FaaS函数会被其他BaaS服务所拓展并且有对象存储或文件存储可以用于长期存储状态信息。开发者可自定义软件服务并且运行相应的应用，从而实现流数据处理、机器学习等功能。通用的Serverless架构可以支持所有运行在服务器上的架构。
- Serverless可应用场景无处不在。在Serverless架构中，通过API网关和函数计算的集成，用户可快速实现API后端服务。通过对象存储和函数计算的事件集成，函数可实时响应对象创建、删除等事件，实现以对象存储为中心的大规模数据处理。通过消息中间件和函数计算的事件集成，用户可快速实现海量消息的处理。无论是一方云服务，还是三方的SaaS服务，或者是用户自建的系统，所有的事件都可快速便捷的被函数计算处理。

Serverless中特定场景与通用场景的对比

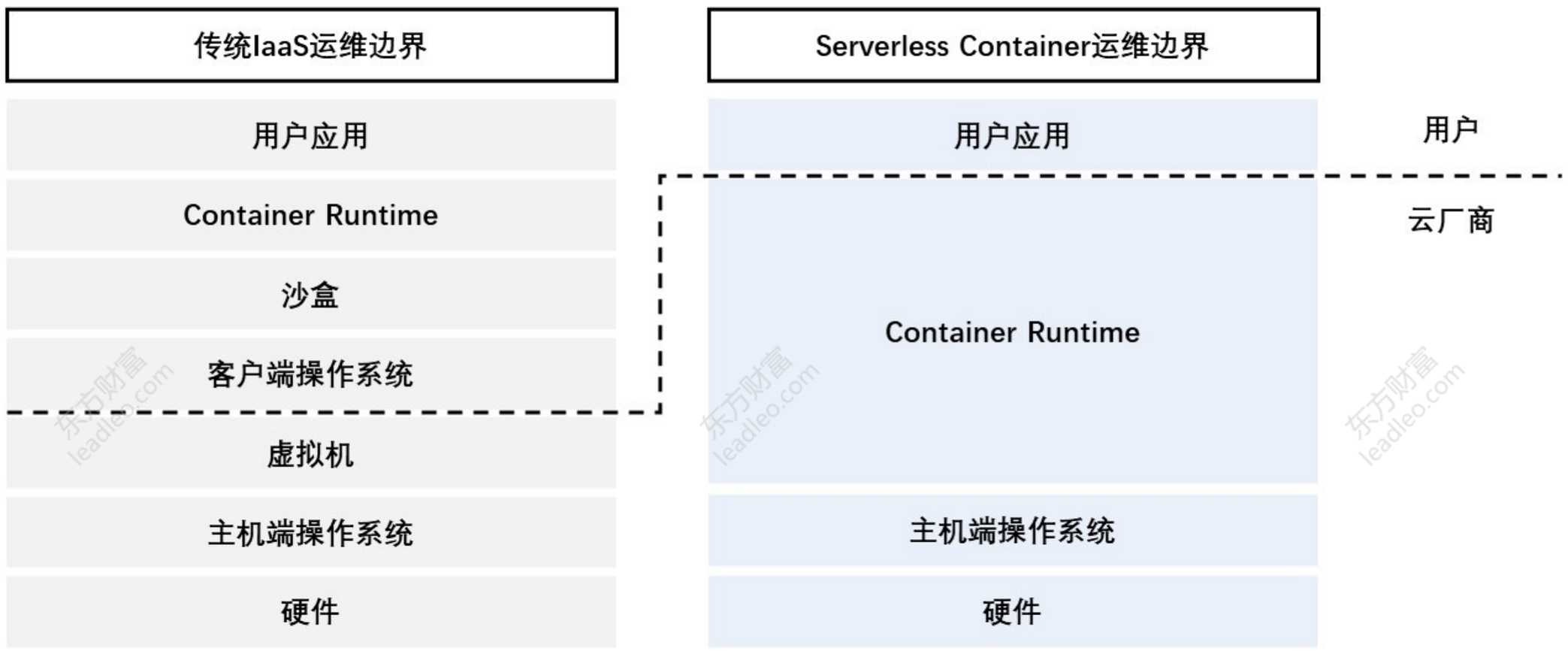


来源：UC Berkley，弗若斯特沙利文

发展趋势二：Serverless与容器深度融合

- 容器具有极强的可移植性，并可为开发者提供敏捷的交付流程。
  - **容器的缺陷：**容器已成为多数应用运行的基础，但开发者仍需负责部分基础设施的管理，因此行业内诞生了Serverless Container服务形式，帮助用户专注于应用构建；
  - **Serverless的缺陷：**Serverless在开发者习惯的兼容性、可移植性、工具链以及生态等方面较为欠缺。
- 容器丰富的工具生态与Serverless免运维、高弹性的性质结合，可以大幅扩展产品功能，提升开发者的开发体验。容器应用Serverless化的典型应用场景包括：
  - **事件驱动音视频处理：**音视频处理流量波动大、且对计算资源弹性要求高，且音视频处理依赖工作流和队列等服务，容器Serverless服务可满足场景需求；
  - **Serverless AI/ML 模型预测、推理服务：**容器镜像可将机器学习框架的示例部署在函数计算上，主流机器学习框架均默认以容器镜像方式进行分享和复用；
  - **传统 Web单体 HTTP 应用 Serverless 演进：**传统Web单体升级具有责任拆分、减轻运维压力以及成本优化的诉求。但传统Web应用业务复杂程度高，运行环境和业务逻辑高度耦合，不能轻易应用Serverless；但基于函数计算的容器镜像帮助传统Web单体无需改造即可使用Serverless服务；
  - **跨云厂商混合部署：**容器镜像是云上、云下的软件交付物统一的默认选择。容器中代码编程方式不与云厂商绑定，可以实现跨云厂商混合部署。
- 未来，云原生的Serverless和容器两大技术的融合程度将加深，开发、部署、运维间的差异将不断缩小，开发者可不需修改业务逻辑即能为不同的工作负载选择合适的技术方案，用开放、标准、统一的云原生技术持续创新。

Serverless容器运维边界



来源：AWS，阿里云，弗若斯特沙利文

发展趋势三：Serverless场景优化

- Serverless根据自身的负载特点，从应用框架、语言、硬件等层面进行端对端优化，例如：
  - 开发新的Java虚拟机技术提高Java应用的启动速度；
  - 通过搭载非易失性内存帮助实例更快被唤醒；
  - 优化CPU硬件与操作系统的协作对高密环境下性能扰动实现精细隔离。

底层硬件优化：异构架构提升算力

- 现阶段云计算大量依赖x86架构，但x86处理器的性能提升速率有限，而AI对算力要求越来越高。Serverless可引入新的架构提升计算效率，如GPU、FPGA、TPU等架构的处理器，让开发者或云服务商自行选择最合适的硬件处理任务，实现底层资源更高的利用率以及更强的服务性能。异构硬件虚拟化、资源池化以及异构资源调度、应用框架的成熟可以提升Serverless架构中Serverful部分的性能，释放更大的商用潜力。

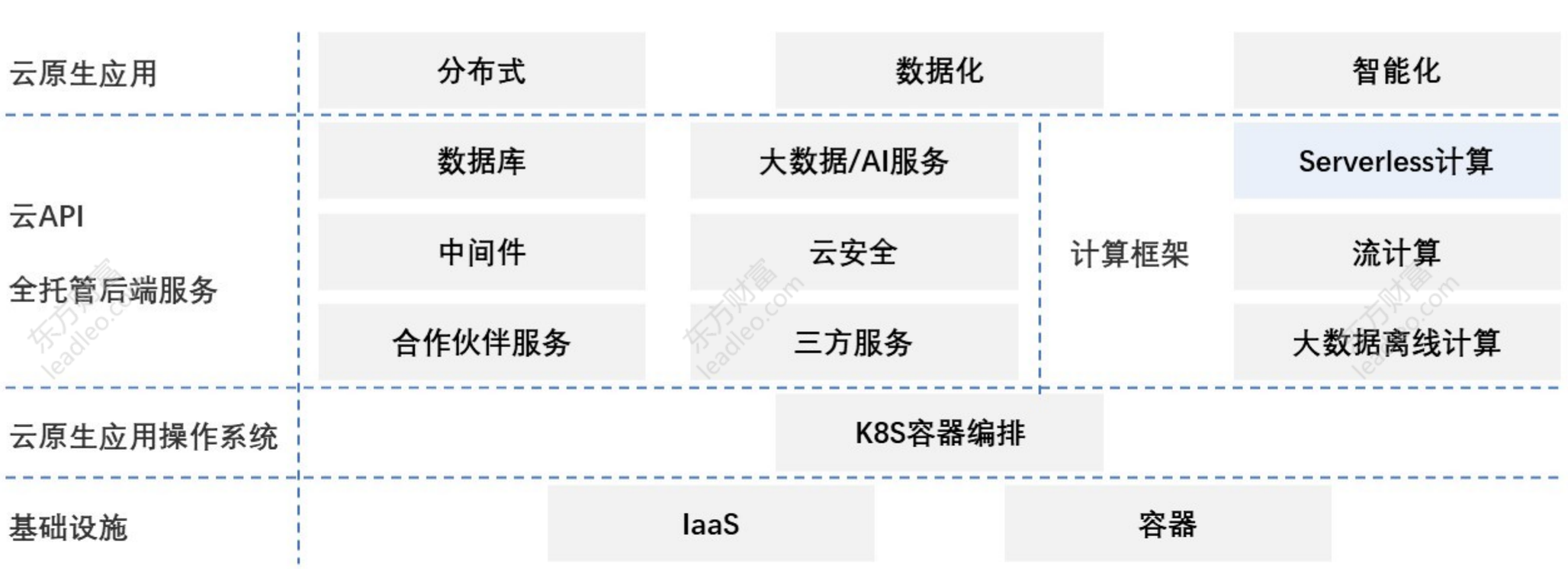
Serverful优化：轻量虚拟化技术提升兼容性

- Serverless的需求包括强安全性、低开销以及对原有程序执行方式的兼容。虚拟机技术具有强安全性与低成本，但虚拟机一般仅适用于特定语言，而容器技术安全性较虚拟机技术差且成本较高，因此AWS、Google等厂商率先研发了轻量虚拟化技术。例如AWS Firecracker通过对设备模型的裁剪和kernel加载流程的优化，实现了百毫秒的启动速度和极小的内存开销。

软件层优化：计算密度持续提升

- 云服务商会针对Serverless 机器学习场景做优化，包括性能、效率和可靠性等方面。云服务商会提供工作流调度、环境配置等能力实现该场景的支持。通过云服务商的上下游能力可以进一步提升Serverless 通用场景下的平台性能。通过一系列的优化，Serverless可持续提升计算密度，实现最佳性能功耗比和性能价格比。

Serverless与云原生生态



来源：阿里云原生，弗若斯特沙利文

名词解释

- **API：**Application Programming Interface，应用程序编程接口。API是预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。
- **云原生：**基于分布部署和统一运管的分布式云，以容器、微服务、DevOps等技术为基础建立的一套云技术产品体系。
- **容器：**容器让开发者可以将应用与库和其他依赖项打包，提供独立环境来运行您的软件服务。
- **虚拟机：**通过软件在一个完全隔离环境中模拟运行、具有完整硬件系统功能的计算机系统。
- **微服务：**一项在云中部署应用和服务的新技术。在微服务架构中，开发者只需要在特定的某种服务中增加所需功能，而不影响整体进程的架构。

东方财富 leadleo.com

方法论

- 头豹研究院布局中国市场，深入研究10大行业，54个垂直行业的市场变化，已经积累了近50万行业研究样本，完成近10,000多个独立的研究咨询项目。

- 研究院依托中国活跃的经济环境，从云原生、无服务器、容器等领域着手，研究内容覆盖整个行业的发展周期，伴随着行业中企业的创立，发展，扩张，到企业走向上市及上市后的成熟期，研究院的各行业研究员探索和评估行业中多变的产业模式，企业的商业模式和运营模式，以专业的视野解读行业的沿革。

- 研究院融合传统与新型的研究方法，采用自主研发的算法，结合行业交叉的大数据，以多元化的调研方法，挖掘定量数据背后的逻辑，分析定性内容背后的观点，客观和真实地阐述行业的现状，前瞻性地预测行业未来的发展趋势，在研究院的每一份研究报告中，完整地呈现行业的过去，现在和未来。

- 研究院密切关注行业发展最新动向，报告内容及数据会随着行业发展、技术革新、竞争格局变化、政策法规颁布、市场调研深入，保持不断更新与优化。

- 研究院秉承匠心研究，砥砺前行的宗旨，从战略的角度分析行业，从执行的层面阅读行业，为每一个行业的报告阅读者提供值得品鉴的研究报告。

东方财富 leadleo.com

法律声明

东方财富 leadleo.com

- 本报告著作权归头豹所有，未经书面许可，任何机构或个人不得以任何形式翻版、复刻、发表或引用。若征得头豹同意进行引用、刊发的，需在允许的范围内使用，并注明出处为“头豹研究院”，且不得对本报告进行任何有悖原意的引用、删节或修改。

- 本报告分析师具有专业研究能力，保证报告数据均来自合法合规渠道，观点产出及数据分析基于分析师对行业的客观理解，本报告不受任何第三方授意或影响。

- 本报告所涉及的观点或信息仅供参考，不构成任何证券或基金投资建议。本报告仅在相关法律许可的情况下发放，并仅为提供信息而发放，概不构成任何广告或证券研究报告。在法律许可的情况下，头豹可能会为报告中提及的企业提供或争取提供投融资或咨询等相关服务。

- 本报告的部分信息来源于公开资料，头豹对该等信息的准确性、完整性或可靠性不做任何保证。本报告所载的资料、意见及推测仅反映头豹于发布本报告当日的判断，过往报告中的描述不应作为日后的表现依据。在不同时期，头豹可发出与本报告所载资料、意见及推测不一致的报告或文章。头豹均不保证本报告所含信息保持在最新状态。同时，头豹对本报告所含信息可在不发出通知的情形下做出修改，读者应当自行关注相应的更新或修改。任何机构或个人应对其利用本报告的数据、分析、研究、部分或者全部内容所进行的一切活动负责并承担该等活动所导致的任何损失或伤害。