

分布式数据库技术系列概览： 分布式数据库核心技术发展趋势

Distributed Database Technology Overview :
Development Trend of Distributed Database Principle

分散データベーステクノロジーの概要：
分散データベースのコア技術の開発動向

报告标签：数据库、云原生、人工智能、开源社区

主笔人：胡俊杰

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，头豹研究院保留采取法律措施、追究相关人员责任的权利。头豹研究院开展的所有商业活动均使用“头豹研究院”或“头豹”的商号、商标，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。

概览摘要

随着云计算的发展，分布式数据库的潜力不断显现，吸引了众多有野心有实力的ICT巨头与独角兽公司。在美国，亚马逊、谷歌等数据库新玩家，正在凭借分布式数据库疯狂侵吞甲骨文的地盘。而在中国，分布式数据库成为了国产化替代弯道超车的最佳路径。

2010年，甲骨文 Oracle 在最权威的数据库性能测试“TPC-C”中跑出史诗级高分30249688，一骑绝尘超出了死对头 IBM 将近 200%，为数年的缠斗画上钢铁般的句号。

2019年，甲骨文中国大规模裁员。同年，蚂蚁 OceanBase跑出了60880800分，时隔九年超越 Oracle 的30249688一倍；华为推出 GaussDB，并成功上线招商银行/工商银行核心系统；中信信用卡系统运行在中兴GoldenDB之上。

2015年成立的开源分布式数据库厂商PingCAP，在2021年5月发布了TiDB 5.0，核心代码（自主可控率）100%，其海外市场营收已经超过了国内营收。

数据库软件原本是卡脖子技术，国产厂商已然通过自主攻关打破了甲骨文与IBM的技术垄断。但由于数据库是一个需要下游配套的复杂系统，国产数据库厂商在破局后最重要的是打造生态，加速布局应用端，稳步出海。

01 分布式事务强一致性

金融核心业务要求数据库必须支持强一致性方案以优先保证数据正确性，保障钱财安全是最重要的应用场景。

从以2PC为代表的写一致性协议过渡到以Paxos为代表的多数派一致性协议，可应用的分布式事务处理技术走向成熟。

分布式数据库以多副本机制来保证系统的安全可靠，而多副本带来的一致性问题的解决则让以Raft为代表的一致性协议成为关键。

02 无共享架构 HTAP混合交易分析型数据库

在高可用性、带宽要求低、平滑扩展能力与远程部署能力上的优势决定了无共享架构将成为未来的存储应用主流。

HTAP解决了金融级数据库面对海量数据库下的存储与并发处理的扩展性问题，实现在线高并发OLTP及OLAP海量数据分析。

03 云化与微服务化 运维上云与智能化

分布式数据库满足云原生需求，在微服务化应用开发及云化平台的趋势下，混合云与私有云部署将成为长期暂态。

运维云化与智能化是适应海量计算环境中保障数据库安全高效得运行的必然发展路线。

目录

◆ 名词解释	-----	08
• 分布式事务强一致性	-----	09
• 无共享架构	-----	14
• HTAP混合交易分析型数据库	-----	16
• 云化与微服务化	-----	18
• 运维上云与智能化	-----	20
• 人工智能赋能数据库	-----	22
• 开源社区	-----	23
◆ 方法论	-----	25
◆ 法律声明	-----	26

CONTENTS

◆ Terms	-----	08
• Consistency of Distributed Transactions	-----	09
• Share-nothing Architecture	-----	14
• HTAP Hybrid Transaction Analysis Database	-----	16
• Cloud Service and Micro-service	-----	18
• Operation and Maintenance	-----	20
• AI for Database	-----	22
• Open source community	-----	23
◆ Methodology	-----	25
◆ Legal Statement	-----	26

■ 图表目录

▪ 三中心跨机房多活部署	-----	09
▪ CAP理论定义	-----	10
▪ 一致性协议简要	-----	10
▪ Raft协议选举原理	-----	11
▪ Raft协议同步原理	-----	11
▪ 一致性协议简要原理	-----	12
• 2PC vs. 3PC vs. Paxos vs. Raft	-----	13
▪ 数据库集群架构分类及比较	-----	14
▪ SN vs. SM vs. SD	-----	15
▪ 传统数据库应用架构数据流	-----	16
▪ 分离架构HTAP简要原理	-----	17
▪ OLTP vs. OLAP vs. HTAP	-----	17
▪ 基础设施的演进	-----	18
▪ 应用架构的演进	-----	18
▪ 分布式改造的演化	-----	19
▪ 运维演变历程	-----	20
▪ 智能运维原理	-----	21
▪ AI赋能数据库对比	-----	22
▪ 开源社区构成	-----	23

名词解释

- ◆ **ACID理论**：是指数据库管理系统（DBMS）在写入或更新资料的过程中，为保证事务（transaction）是正确可靠的，所必须具备的四个特性：原子性（atomicity，或称不可分割性）、一致性（consistency）、隔离性（isolation，又称独立性）、持久性（durability）
- ◆ **BASE理论**：通过牺牲强一致性来获得可用性，并允许数据段时间内的一致，但是最终达到一致状态。BASE是三个短语的缩写：基本可用（Basically Available），软状态（Soft State），最终一致性（Eventually Consistent）。
- ◆ **GOSSIP协议**：又称 epidemic 协议(epidemic protocol),是基于流行病传播方式的节点或者进程之间信息交换的协议,在分布式系统中被广泛使用。
- ◆ **大规模多级并行处理能力MPP**：massively parallel processing，是采用大量处理单元对问题进行求解的一种并行处理技术。
- ◆ **MVC架构**：M是指业务模型，V是指用户界面，C则是控制器，使用MVC的目的是将M和V的实现代码分离，从而使同一个程序可以使用不同的表现形式。
- ◆ **SOA架构**：面向服务的架构（SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）进行拆分，并通过这些服务之间定义良好的接口和协议联系起来。
- ◆ **微服务架构**：微服务架构 = 80% 的 SOA 服务架构思想 + 100% 的组件化架构思想 + 80% 的领域建模思想，系统中的各个微服务可被独立部署，各个微服务之间是松耦合的。每个微服务仅关注于完成一件任务并很好地完成该任务。
- ◆ **DevOps**：Development和Operations的组合同，是一组过程、方法与系统的统称，用于促进开发（应用程序/软件工程）、技术运营和质量保障（QA）部门之间的沟通、协作与整合。
- ◆ **AIOps**：是应用人工智能 (AI) 包括大数据、分析和机器学习的功能来改进 IT 运营的方法。
- ◆ **NoOps**：使开发、部署和维护，以便可以将人从循环中移除，一切都得到了处理。
- ◆ **LXC轻量级容器**：Linux Container容器是一种内核虚拟化技术，可以提供轻量级的虚拟化，以便隔离进程和资源。
- ◆ **Zabbix**：一个基于WEB界面的提供分布式系统监视以及网络监视功能的企业级的开源解决方案。能监视各种网络参数，保证服务器系统的安全运营；并提供灵活的通知机制以让系统管理员快速定位/解决存在的各种问题。
- ◆ **数字孪生**：数字孪生是充分利用物理模型、传感器更新、运行历史等数据，集成多学科、多物理量、多尺度、多概率的仿真过程，在虚拟空间中完成映射，从而反映相对应的实体装备的全生命周期过程。
- ◆ **APM应用性能管理**：Application Performance Management，是一个比较新的网络管理方向，主要指对企业的业务应用进行监测、优化，提高企业应用的可靠性和质量，保证用户得到良好的服务，降低IT总拥有成本。
- ◆ **CMDB配置管理数据库**：Configuration Management Database，配置管理数据库包含一个组织的IT服务使用的信息系统的组件的所有相关信息以及这些组件之间的关系。

分布式事务一致性（一）

- 分布式数据库不同于单体数据库，必须考虑一致性方案以保证不同节点中的数据一致与完整

一致性的定义

在数据库系统中通常用事务来保证数据的一致性和完整性。在分布式系统中，数据一致性往往指的是由于数据的复制，不同数据节点中的数据内容是否完整并且相同。

弱一致性：异步复制，不能保证任何一次读都能读到最近一次写入的数据。

强一致性：同步复制，任何服务器上的读操作总是能看到最新写入的数据。

不同的业务需求与环境对一致性的要求不一样，金融级需求强调强一致性以保证钱财的安全与一致；互联网需求强调高可用可牺牲一定的一致性以达到性能优先。

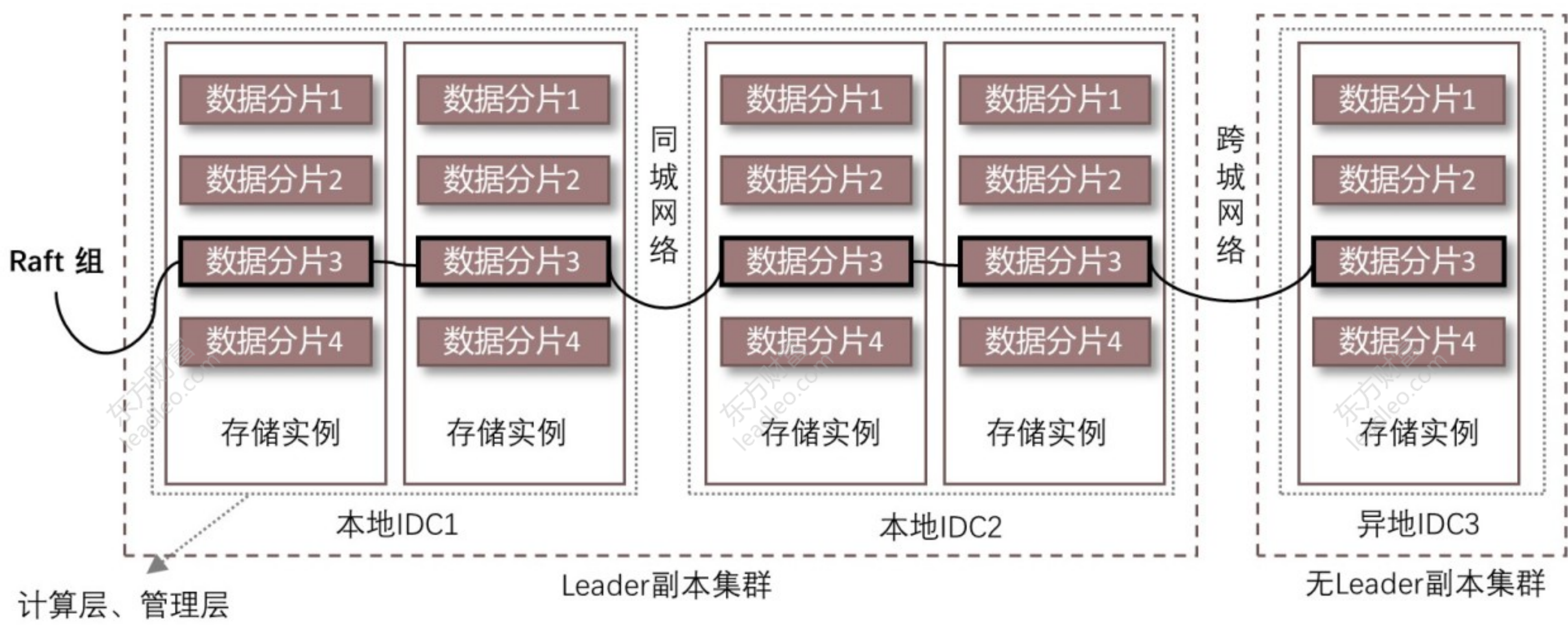
实现强一致性的底层路径

写一致性：在写入数据的同时将数据复制到其他服务器上；

读一致性：写入后不再复制，在读取时使用版本来协调复制；

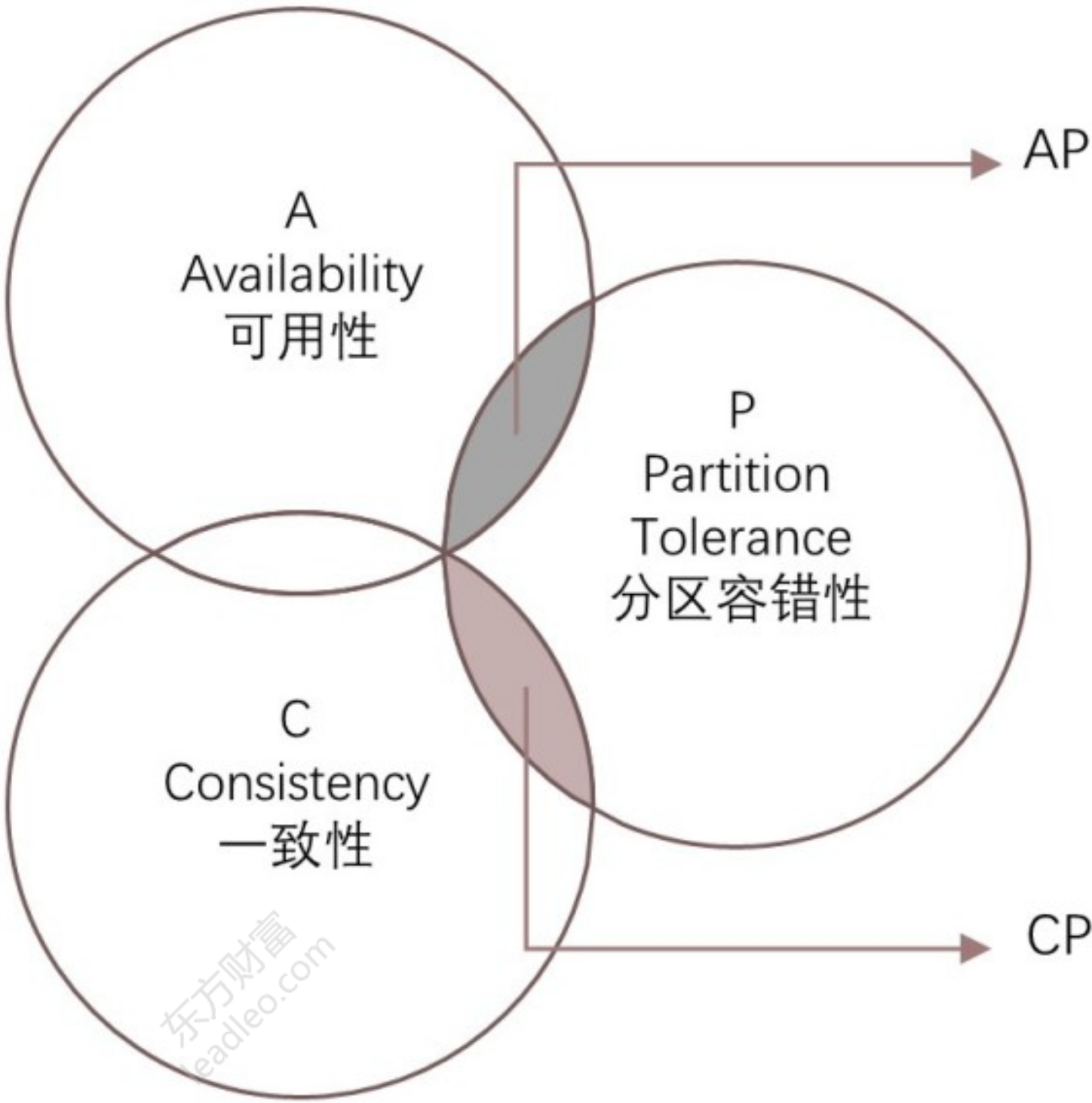
多数派一致性：引入数据版本号判断并读取最新写入的多数派数据副本。

三中心跨机房多活部署



来源：TiDB，沙利文

CAP理论定义



来源：头豹研究院

CAP理论

CAP理论证明了，任何分布式系统只可同时满足以上两点，无法完全兼顾三者。而分布式系统都需要满足分区容错性，必须在一致性和可用性之间进行权衡。

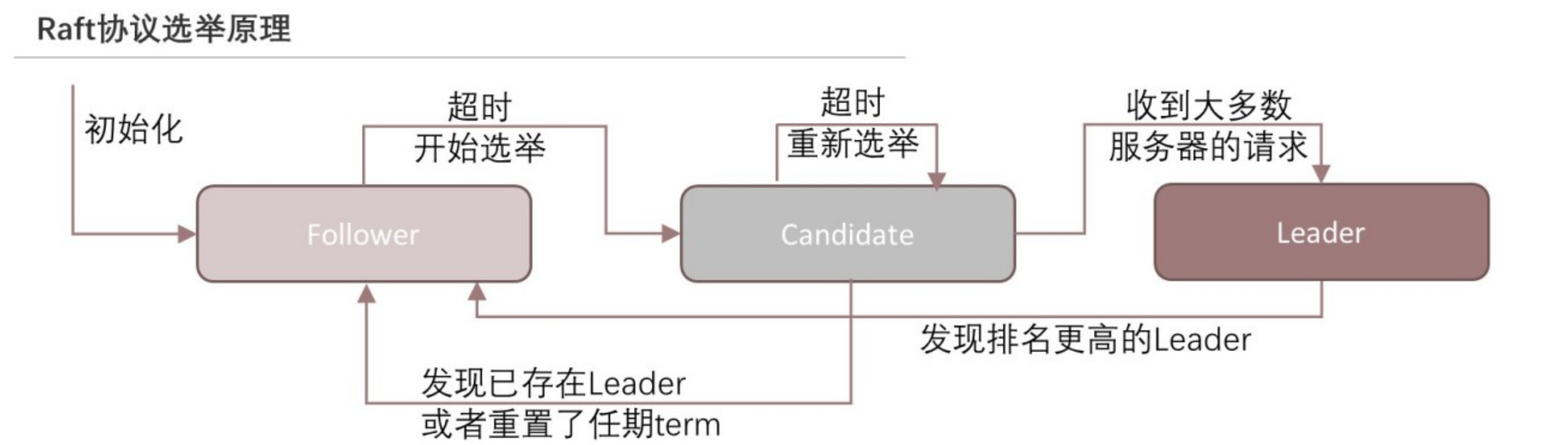
- AP系统：选择了高可用和分区容错性，此时，那个失去联系的节点依然可以向系统提供服务，不过它的数据就不能保证是实时同步的。
- CP系统：要求数据在各个服务器上**强一致**的，然而网络分区会导致同步时间延长，可用性得不到保障。适合对结果一致性非常敏感的金融级数据库，保障钱财安全是最大的应用场景。

一致性协议简要

弱 一 致 性	BASE理论	牺牲部分一致性换取高可用性
	GOSSIP协议	随机传播，最终一致性
强 一 致 性	主从同步复制	主机完成读写后同步至从机
	二阶段提交协议	协调参与者共同达到提交/回滚的结果
	三阶段提交协议	解决二阶段协议宕机情况阻塞问题
	Paxos协议	收到半数以上的最大版本的提案后，将所有节点写入同一个值
	Raft协议	大幅简化Paxos协议且功能等价

分布式事务强一致性（二）

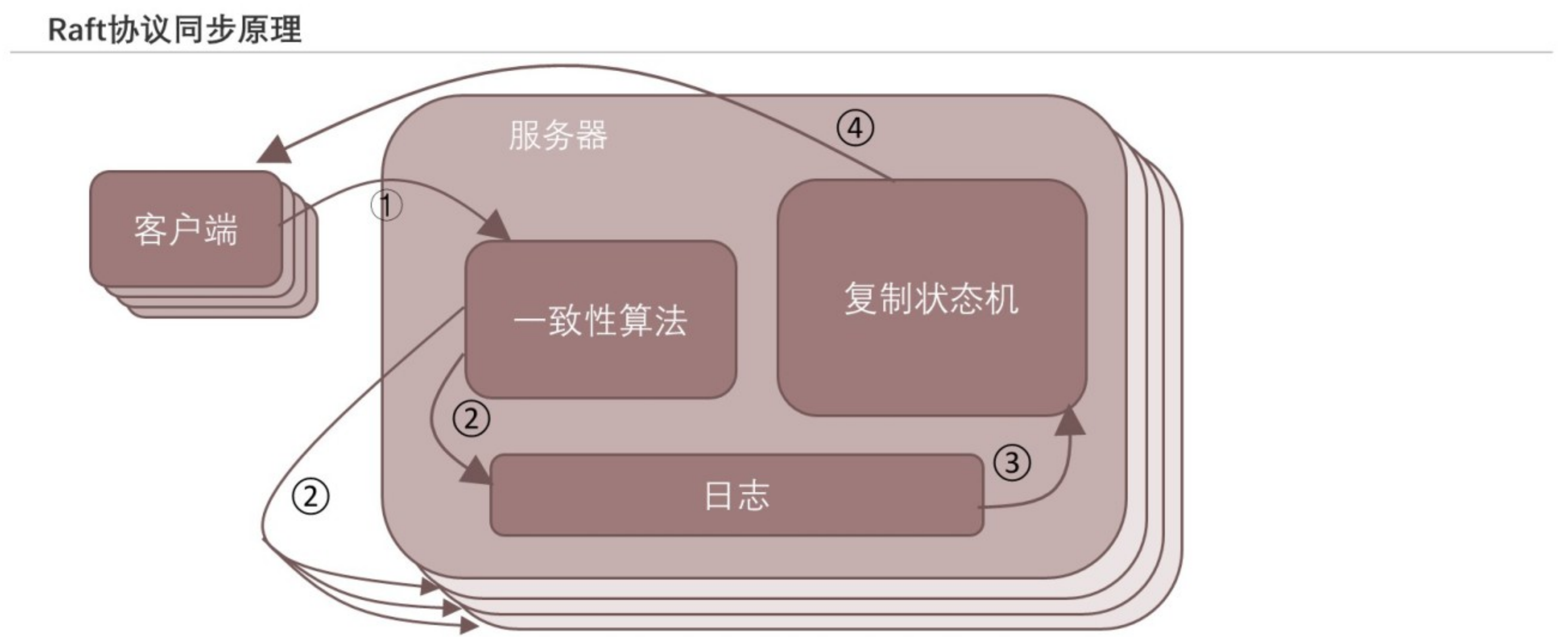
- 分布式数据库以多副本机制来保证系统的安全可靠，而多副本带来的一致性问题则让以Raft为代表的 consistency 协议成为关键



来源：斯坦福大学，头豹研究院

Raft算法中服务器主要分为三种角色，并且三种角色相互独立，在同一时间内只可能扮演其中一种角色。

- Leader：用于对所有用户的请求进行处理以及日志的复制等等。
- Follower：不会主动发送消息，只响应来自Leader与Candidate的请求。
- Candidate：用于选举新的Leader。



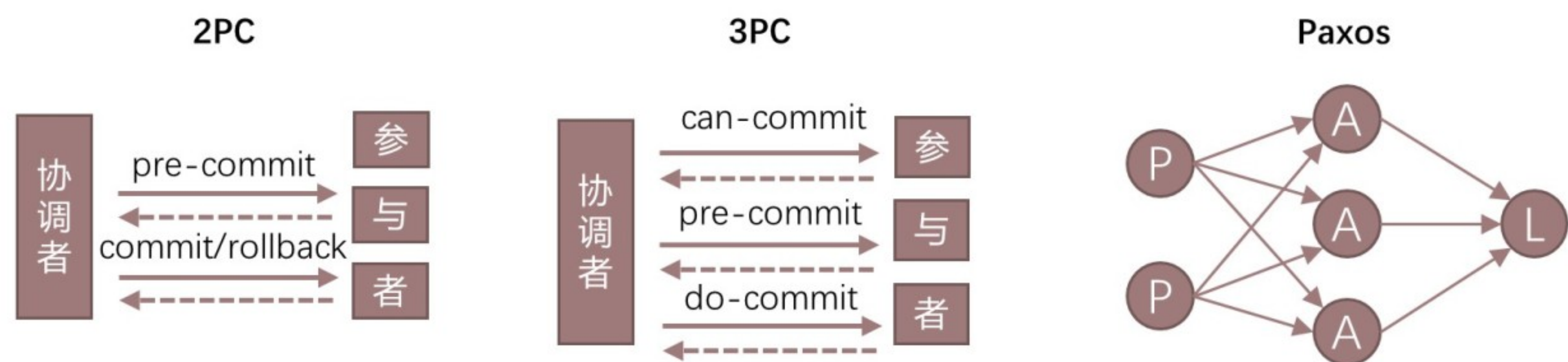
来源：斯坦福大学，头豹研究院

- 客户端发起请求，Raft一致性算法将请求交由Leader处理
 - Leader将操作写入日志，紧接着对Follower发起追加请求，尝试让Follower更新日志副本
 - 如果Follower过半数同意请求，Leader进行commit操作，把指令交由状态机处理
 - 状态机处理完成后，返回结果给客户端。
- 一致性算法管理来自客户端状态命令的复制日志，保证状态机处理的日志中的命令的顺序都是一致的，因此会得到相同的执行结果。

分布式事务强一致性（三）

- 从以2PC为代表的写一致性协议过渡到以Paxos为代表的多数派一致性协议，可应用的分布式事务处理技术走向成熟

三种一致性协议简要原理



2PC 两阶段提交协议

2PC是一种原理比较简单的分布式一致性协议，由一个中心化协调者节点对N个参与者节点组成，事务的提交过程分成了两个阶段来进行处理。

优点：原理简单，方便实现强一致性。

缺点：阻塞性协议，单点协调者服务宕机会严重影响系统的可用性。

案例：TiDB、GoldenDB、GaussDB

3PC 三阶段提交协议

将 2PC 的Pre-Commit过程一分为二(Can-Commit、Pre-Commit)

优点：降低了阻塞，缓解了单点故障问题。

缺点：没有摆脱中心化单点服务，仍存在不一致性隐患，性能较慢。

案例：N/A

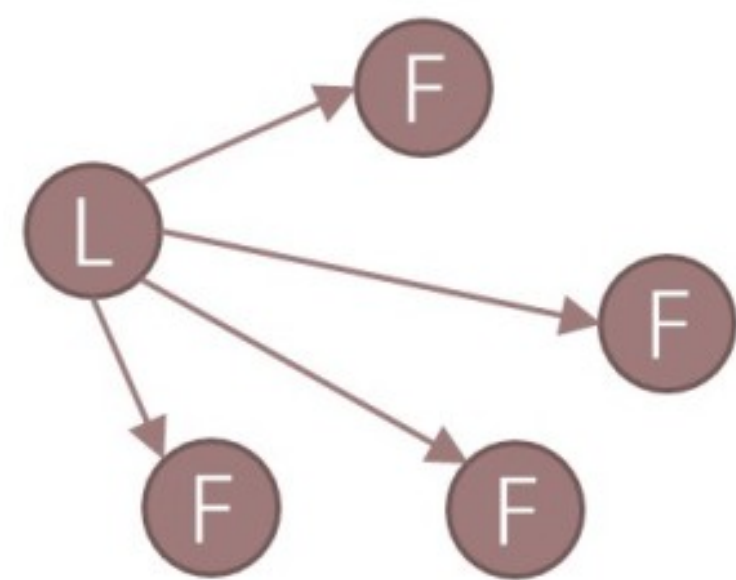
Paxos协议

基于消息传递且具有高度容错特性的共识算法，一个分布式系统中的各个进程如何就某个值（决议）通过共识达成一致。Paxos将系统中的角色分为Proposer，Acceptor和Learner，一个进程可能同时充当多种角色，当一半以上的Acceptor同意了一个提案，就认为该提案被选定了。

优点：彻底避免了单点故障，支持多节点同时写入，高并发度。

缺点：算法实现难度和运行开销非常大，算法陷入活锁的情况没有解决。

案例：OceanBase、GaussDB、PolarDB



❑ Raft协议

Raft协议是Paxos协议的一种简化实现， Raft将系统中的角色分为Leader, Candidate和Follower。Raft强化了节点间的日志同步，使得选举后的日志同步更加清晰，在效果上与Paxos等价。

优点：实现与运行更简单，多场景适用，单节点写入保证更强的一致性。

缺点：同步日志机制降低了新加入节点性能，单节点投票机制降低并发度。

案例：TiDB、RadonDB、DM8、GoldenDB

2PC vs. 3PC vs. Paxos vs. Raft

	数据一致性	安全性	容错性	性能扩展性	高并发性能	运维成本
2PC	2	1	1	1	2	1
3PC	2	1+	1	1	1	2
Paxos	3	2	1	1	3	1
Raft	3+	3	3+	2	2	1



扫码查看高清图片

<https://www.leadleo.com/table/details?id=60f647db9b8e8002cd9d97bc&core=610bb21adbb9b0070765af64>

来源：51CTO，上海交通大学，纽约大学，头豹研究院

❑ 四种架构的比较

Raft协议在保证数据一致性、安全性、容错性上都有着明显的优势，在匹配金融级需求上更贴合；

Paxos协议在高并发性能支持上有着明显优势，且能保证不俗的一致性和容错性，使其在匹配互联网需求上更贴合。但Paxos的设计原理极为复杂，工业级Paxos实现上各个厂商都有着不同的理解和实施方案，使得其维护成本也更高。

2PC二阶段提交在低复杂性和低维护成本有着明显优势，但容错性较低，在发生宕机时有数据丢失的风险。

无共享架构

- 在高可用性、带宽要求低、平滑扩展能力与远程部署能力上的优势决定了无共享架构将成为未来的存储应用主流

数据库集群架构分类及比较



来源: Oracle, ScaleDB,, 韩国工业科学院RIST, 加州大学伯克利分校, 头豹研究院

SN vs. SM vs. SD

	并行控制能力	灾难恢复能力	数据库设计易度	负载均衡能力	高可用性	消息数量处理	带宽要求	横向扩展能力	远程部署能力	关键会话敏感性	系统镜像数量	热点敏感度
SN	2	2	1	1	3	1	3	3	3	3	1	1
SM	2	3	2	3	1	3	1	1	1	1	3	1
SD	1	1	2	2	2	2	2	2	2	2	1	1

1-3 为推荐分，更推荐得分高者
来源：加州大学伯克利分校，头豹研究院

三种架构的比较

共享内存架构无法大规模扩容处理器，使得此架构相较于其他二者失去了最为重要的竞争优势，扩展能力决定了其集群性能的上限；

共享磁盘架构没有在任何一项能力之中胜出（所有得分都为1或2）；

尽管无共享架构存在难以对付的缺陷，但其在高可用性、带宽要求低、扩展能力与远程部署能力上的优势决定了无共享架构将成为未来的应用主流。

无共享架构设计思路深化

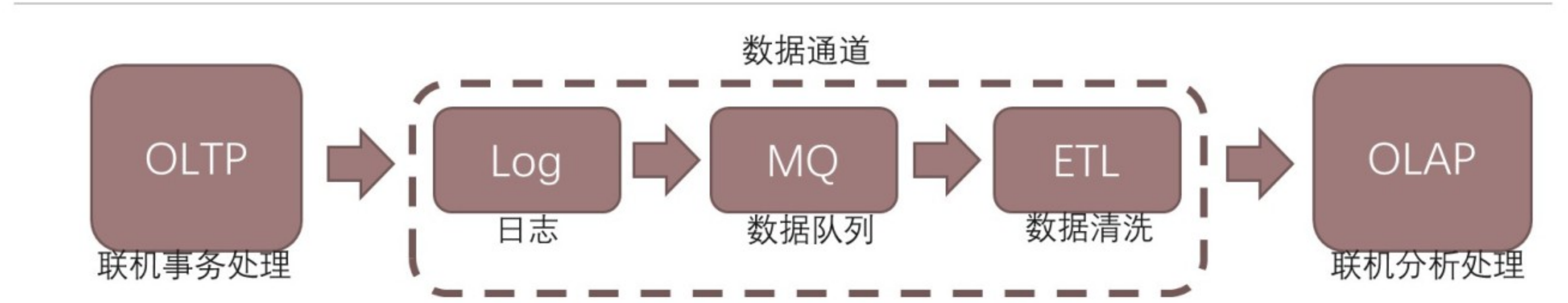
对于共享内存架构与共享磁盘架构而言，扩展更多的处理器，反而会使系统减慢，因为增加了对内存访问和网络带宽的竞争，由于内存系统和磁盘系统都由内部通讯机制联系在一起，而出现了明显的瓶颈。

当随着事务数量不断增加，无共享架构通过增加额外的处理器和内存，降低了竞争资源的等待时间，从而提高了性能，保证了每个事务处理时间不变。反过来，如果一个数据库应用系统要获得良好的可扩展的性能，它从设计和优化上就要考虑无共享体系结构

HTAP混合交易分析型数据库

- HTAP解决了金融级数据库面对海量数据库下的存储与并发处理的扩展性问题，实现在线高并发OLTP及OLAP海量数据分析

传统数据库应用架构数据流



来源：卡内基梅隆大学，PingCAP，头豹研究院

❑ 传统数据流构成

- OLTP主要用来记录某类业务事件的发生，如购买行为，当行为产生后，系统会记录是谁在何时何地做了何事，这样的一行（或多行）数据会以增删改的方式在数据库中进行数据的更新处理操作，要求实时性高、稳定性强、确保数据及时更新成功。
- 当数据积累到一定的量后，根据业务分析需求从分散的业务系统中提取不同的业务数据，做对应的数据清洗后存储在数据仓库中，然后由数据仓库来统一提供OLAP分析。

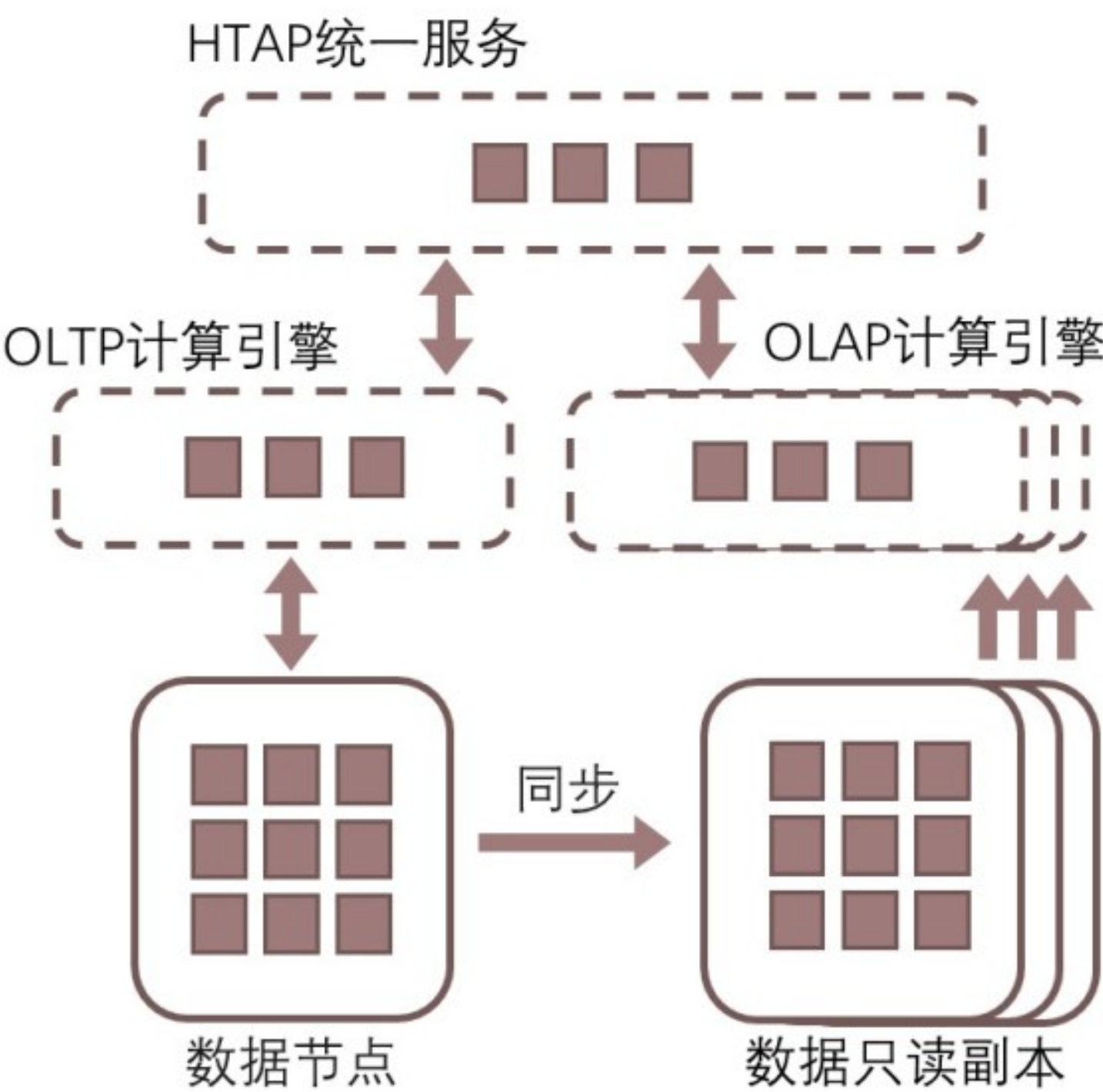
❑ 传统数据库应用架构存储属性

- OLTP**：将不同属性连续存储，即行存储，一条数据的所有属性是连续存储的，可以使得插入/更新/删除更快。
- OLAP**：将不同数据的同一属性连续存储，即列存储，使得查询操作只读关心的数据属性，而不是一整条数据，减少浪费；可以更好地支持复杂查询。

❑ 传统数据库应用架构的问题

- OLAP 和 OLTP 系统间会有分钟/小时级的时延，且OLAP和OLTP数据库之间的一致性无法保证，难以满足对分析的实时性要求很高的业务场景。
- 高可用冗余的副本数量大且分散，ETL数据异步传输链路运维复杂度高，副本同步和运维的难度和成本高。
- 软件开发团队需要为不同的数据库编写查询语句，且有可能需要将不同系统的数据进行聚合，开发成本高。

分离架构HTAP简要原理



来源：GBASE, PolarDB-X, 头豹研究院

HTAP定义与分类

根据OLTP和OLAP负载是否使用相同的节点或引擎，将HTAP分为统一架构和分离架构。

HTAP避免了复杂昂贵的抽取、转换、加载操作（ETL），实现快速分析数据的能力。

分离架构的HTAP实现收敛技术栈

低成本，底层数据可快速复制，并且同时满足高并发的实时更新。

块连续存储，行存储频繁被修改的热数据，列存储需要查询和分析的冷数据。

大规模多级并行处理能力MPP，以无共享架构集群具备线性扩展能力。

资源隔离，提供AP、TP资源链隔离机制，避免相互影响。

HTAP数据库案例

GoldenDB、GaussDB、TiDB、PolarDB、Oceanbase

OLTP vs. OLAP vs. HTAP

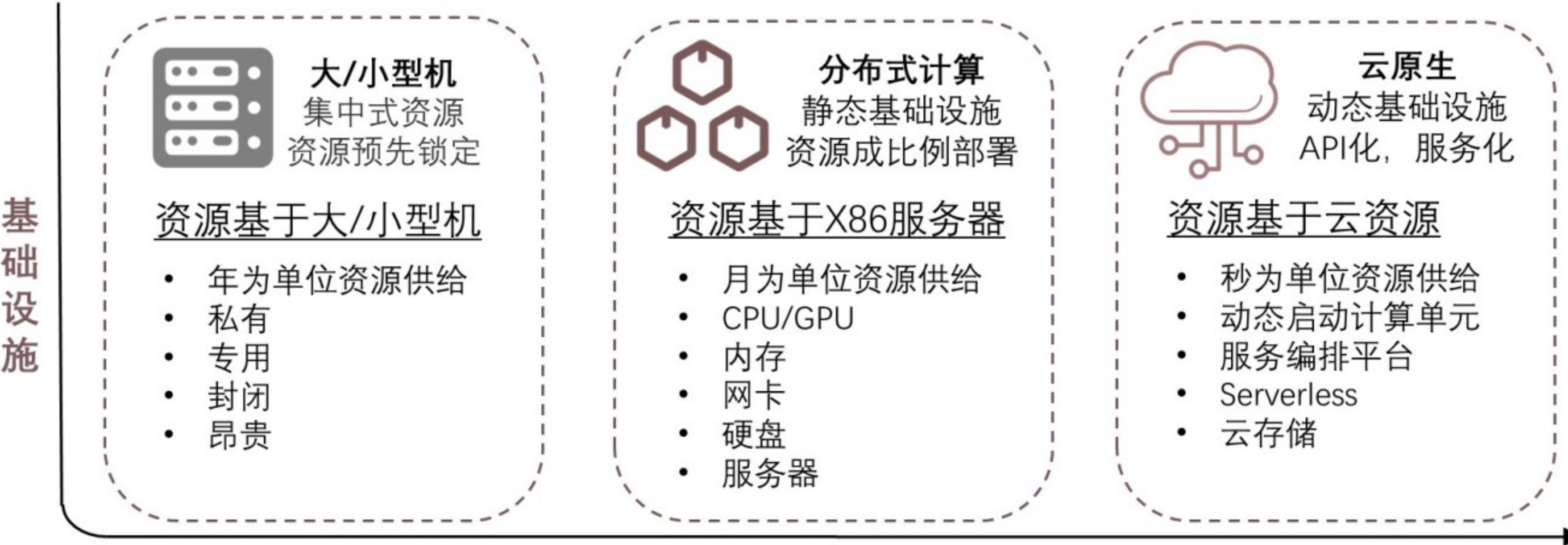
	响应速度	吞吐量	并发访问量	事务大小	数据访问量	索引规模	实时性	物化视图
OLTP	3	1	2	1	1	2	3	1
OLAP	2	2	1	2	2	3	1	3
HTAP	3	3	3	3	3	3	2	3

1-3 为推荐分，更推荐得分高者
来源：卡内基梅隆大学，头豹研究院

■ 云化与微服务化

- 分布式数据库满足云原生需求，在微服务化应用开发及云化平台的趋势下，混合云与私有云部署将成为长期暂态

基础设施的演进

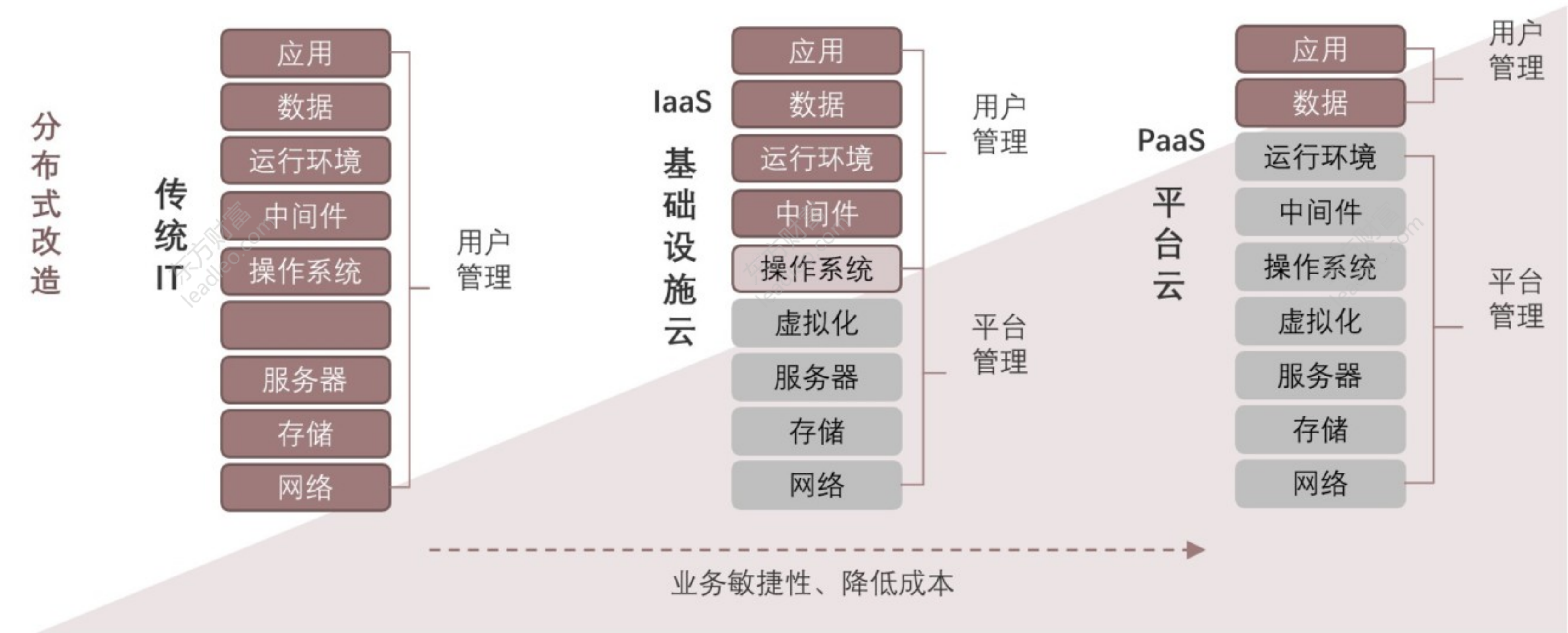


云化趋势

在微服务化应用开发以及云化平台的趋势下，云分布式数据库调用云基础设施资源将不存在限制，满足了上层应用的弹性扩展、高并发、高吞吐量与灵活敏捷的要求。

在中国的金融级需求中，处于核心业务对于数据主权、安全隐私的考量，混合云与私有云部署已成为金融级需求上云的长期暂态。

分布式改造的演化



来源：中兴，头豹研究院

云分布式数据库成为新增长点

当云原生使用微服务、容器等技术，赋能数据库更加敏捷的服务支持和更加弹性的扩展及持续的交互；当资源在云基础设施中的调动不受限制，数据处理和分析的服务不再有性能瓶颈和业务可用性的妥协，每个节点既可以管理本地的数据应用也参与全局数据应用，从而构成一个逻辑上统一的数据库系统。将掀起一场数据库领域的代际变革和能力爆发。

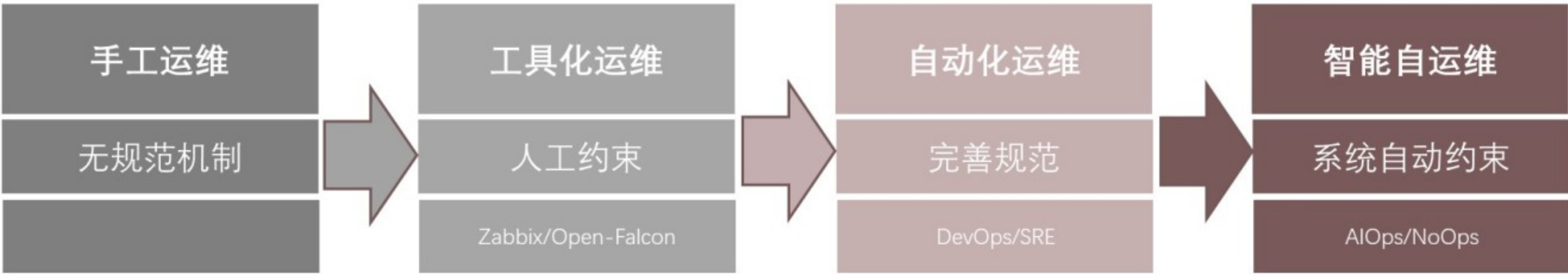
云分布式数据库案例

TiDB、GoldenDB、GaussDB、Oceanbase、PolarDB-X

运维上云与智能化

- 运维云化与智能化是适应海量计算环境中保障数据库安全高效得运行的必然发展路线

运维演变历程



来源：运维派、阿里云、英特尔、中国互联网协会、头豹研究院

传统IT运维

在手工运维和工具化运维的阶段中，运维服务是由技术驱动的，服务形式单一，多以机房或数据中心的IT基础架构为范围，实行响应式、救火式的被动工作模式，自动化程度低，人员成本高。

生产力工具适配生产力发展

用户增长与企业投入系统建设维护的成本呈非线性关系，仅增加运维人员的数量无法满足系统发展需求。为了提高IT的整体效率与质量，总结重复、可追溯的现象，形成规则，完成自动运行维护成为发展的必要。

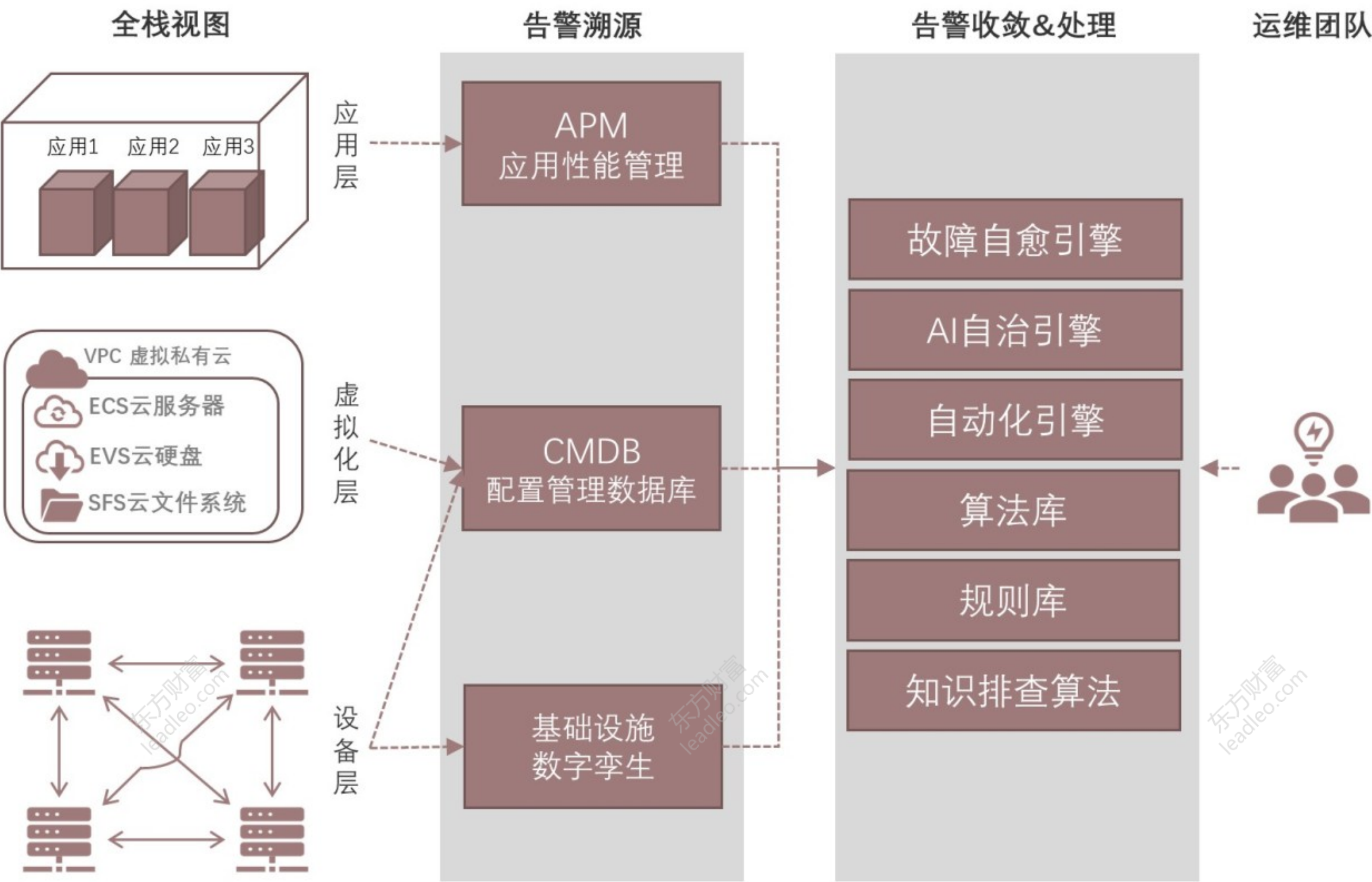
推广Agile敏捷和DevOps

过去的IT团队一般是以定期成立和解散的短期工作组的状态分别实现开发周期中的构建、测试、发布、运维。推广Agile和DevOps实践，让分散的IT子团队能够在对开发周期全局的统一规划指导下，确定并符合开发架构标准，消除不必要的障碍，缩短开发周期，减轻后期的运维压力。

全栈视图

过去的运维工作的困难点在于排障，往往只能在应用层看到告警的发生，但是实际故障点在设备层，对于运维人员而言只能从应用层到虚拟化层再到设备层进行排除，才能发现故障点。而通过全栈视图，可以直达故障点，节省大量的时间和人力。

智能运维简要原理



来源：华为云、阿里云, 头豹研究院

❑ 技术欠账

“技术欠账(Technical Debt)”是IT运维团队在代码、技术文档、开发环境、第三方工具、应用冗余和开发实践方面，阻碍变革、降低效率的不足之处。技术欠账可以借助上云实现清晰化，在云上运维中把AI嵌入最关键的业务流程，AIOps通过自动收集和分析基础架构、应用程序和云服务等指标，以及日志和事件等原始数据，通过机器学习分析并从技术角度识别问题。

❑ 云运维

运维上云是由客户需求驱动服务，服务形式从机房拓展到桌面端、移动端、云端，实行主动式、自动化的工作模式，人员成本低。对比传统IT运维，极大降低了对运维人员的依赖，从故障后处理进阶到故障前规避。

❑ 具备云化与智能化运维的数据库案例

GoldenDB、GaussDB、PolarDB-X

人工智能赋能数据库

- AI算法替代启发式算法，通过学习历史数据，根据现状在运行时进行动态调整，解决传统数据库的痛点

传统数据库

在传统数据库中，大量地使用启发式算法通过传统经验去优化数据库，已然无法满足更高的执行要求，比如针对众多用户的实际场景进行定制化开发。过去采用系统预定义参数组合或可调节参数开关等方式，由DBA根据经验进行调整。

AI for DB（AI4DB）

利用AI算法替代启发式算法，解决传统数据库的痛点，典型方向有：

- 优化器：**传统代价优化基于采样统计信息进行表连接规划，存在统计信息不准、启发是连接规划等问题。
- 参数调优：**数据库有数十至上百个可调节参数，其中很多参数时连续值调节空间，依靠人工经验无法找到最优参数组合。
- 自动化索引推荐和视图推荐：**在数据库的众多表与列中自动构建索引和视图，来提升数据库的性能。
- 事务智能调度：**事务的并发冲突时OLTP的难点，可以通过人工智能进行智能调度来提升数据库的并发性能。

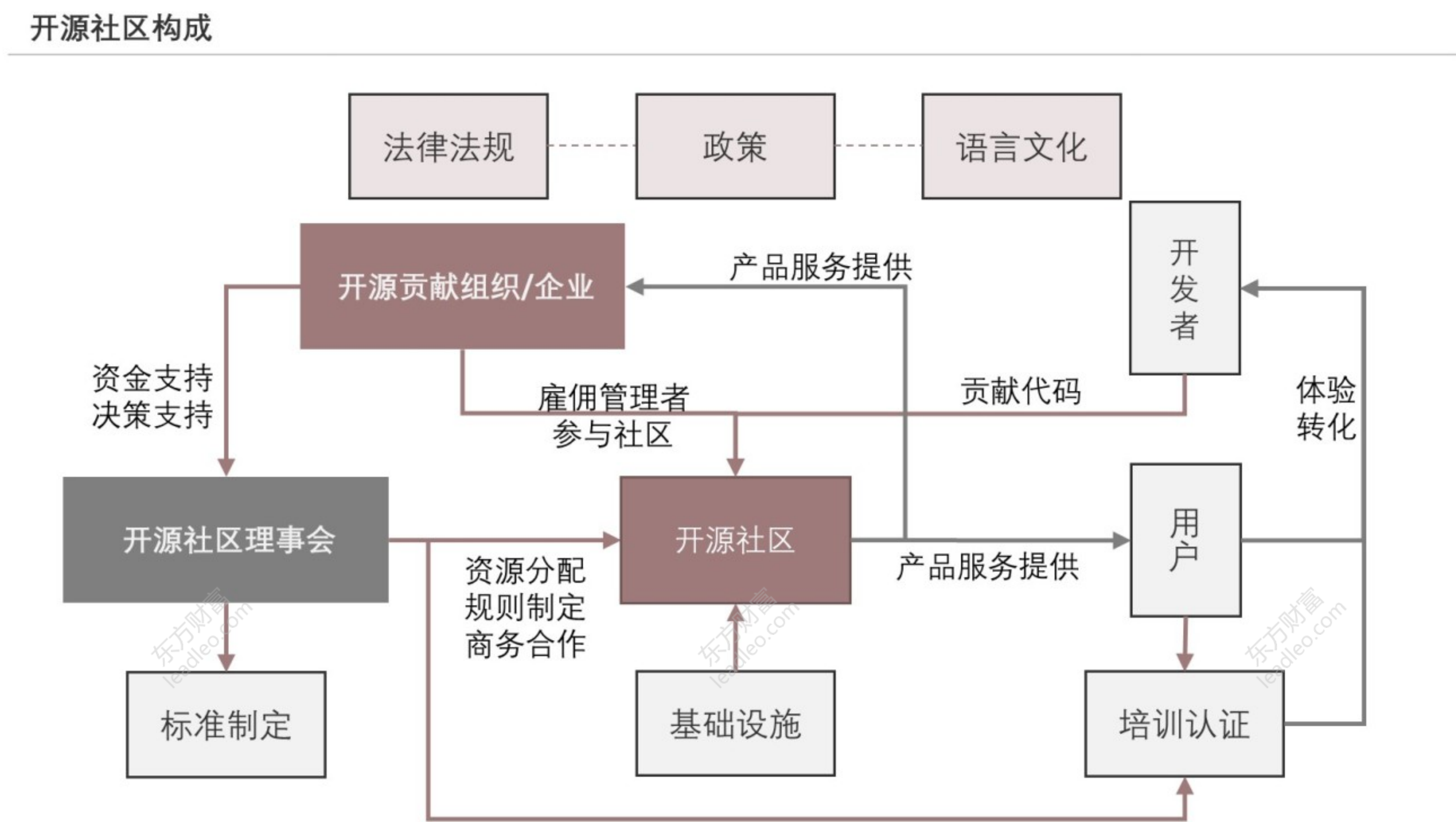
AI赋能数据库对比

		工 作 适 用 性					数 据 场 景					
		调 时 优 长	训 间 练 成 时 本	质 量	负 载	应 性	硬 件	适 用 性	数 据 场 景	质 量	适 用 性	
传统机器学习	人工调优	数日-数周	数月 (训练DBA)	低	X		X		线下索引选择 线上索引选择 半自动索引选择 基于机器学习的索引选择	静态	低	低
	基于搜索	数小时	数小时	低	X		X			动态	中	中
		数秒	数小时	中+	✓		X			动态	高	中
	强化学习	毫秒	数小时-数日	高	✓		✓			动态	高	高
	深度学习	毫秒	数小时-数日	中	✓		✓			动态	高	高

来源：清华大学、头豹研究院

开源生态

- 开源是发展开发者群体和用户群体以及放大技术影响力的最佳选择，是生态推进的重要手段



来源：头豹研究院

开源生态吸引产业链上下游协作

开源数据库能够吸引包括ISV独立软件应用开发商、中间件开发商、OS操作系统开发商、服务器/底层硬件提供商的产业链上下游厂商参与，通过免费提供开源版本，企业获取了用户体验的验证，也形成了对外部开发者持续的吸引力。

开源数据库需求侧

取消技术绑定，降低迁移成本

企业端用户在使用开源数据库作为数据库支撑，可以免除被闭源系统的技术绑定，在开源社区内实现数据库迁移，进行企业需求个性化定制，实现系统间的兼容，解决业务连续性。

标准制定

通过开源确保技术趋于成为数据库行业的主流技术方案。参与国家部门和不同行业监管机构的数据库执行标准的制定，提升自身产品服务能力并符合执行标准，提高用户认可度。

❑ 开源数据库供给侧

开源数据库体系架构相较于封闭式的数据库系统能吸引更多开发者参与。

• 流行度

随着开源数据库的流行程度上升，人才供给将源源不断地支持基于开源数据库的不同开发项目，从而进一步扩大开源数据库的影响力，同时加速开源版本的迭代演进。

• 高校合作

通过产、学、研、用，打通人才体系建设，通过基础理论对接产业和市场需求构建知识体系，比如openGauss已成为国家计算机等级考试的科目内容，构建良性人才生态，激发产业持续活力。

• 培训认证

面向数据库从业者、架构师、开发人员和运营职位的人员打造的基于角色的技能认证，验证数据库领域技术技能和专业知识的，让求职者的职业发展与公司业务发展相匹配，从理论到实践到应用相结合，减少入职学习成本和新人上岗的培训成本，从而绑定人才与企业对本开源数据库的认可度，进而提高开源数据库的可持续发展性。

❑ 外部环境

- 法律法规：开源社区必须满足所处地区相应的法律法规。
- 政策引导：信创大环境下，数据库的软硬件国产化替代加速，开源是推动技术创新实现可用、好用、推广替代的重要手段。
- 语言文化：国产开源社区积累的大量中英文文档资源对海外熟悉中文的代码贡献人和DBA用户等华人更有吸引力。

❑ 盈利模式

MySQL是主流的开源数据库架构，被Oracle收购并控股，Oracle同时作为全球领先的数据库厂商正控制着MySQL的竞争力，稳固自家付费产品和服务的地位。相似的，开源数据库厂商大都保留了具备企业级特性的分布式架构内容转而以付费项目根据具体需求提供更优质的服务，在开源社区中实现提供商和用户的双赢。

❑ 开源数据库案例

openGauss、TiDB、Oceanbase、PolarDB、MySQL、PostgreSQL

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，从社会保险、人工智能、大数据等领域着手，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。

法律声明

- ◆ 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。

■ 头豹研究院简介

- ◆ 头豹研究院是中国大陆地区首家B2B模式人工智能技术的互联网商业咨询平台，已形成集行业研究、政企咨询、产业规划、会展会议行业服务等业务为一体的一站式行业服务体系，整合多方资源，致力于为用户提供最专业、最完整、最省时的行业和企业数据库服务，帮助用户实现知识共建，产权共享
- ◆ 公司致力于以优质商业资源共享为基础，利用大数据、区块链和人工智能等技术，围绕产业焦点、热点问题，基于丰富案例和海量数据，通过开放合作的研究平台，汇集各界智慧，推动产业健康、有序、可持续发展



四大核心服务

- | | |
|--|---|
| <h3>企业服务</h3> <p>为企业提供定制化报告服务、管理咨询、战略调整等服务</p> | <h3>云研究院服务</h3> <p>提供行业分析师外派驻场服务，平台数据库、报告库及内部研究团队提供技术支持服务</p> |
| <h3>行业排名、展会宣传</h3> <p>行业峰会策划、奖项评选、行业白皮书等服务</p> | <h3>园区规划、产业规划</h3> <p>地方产业规划，园区企业孵化服务</p> |

报告阅读渠道

- ◆ 1、头豹科技创新网 —— www.leadleo.com PC端阅读全行业、千本研报



- ◆ 2、头豹小程序 —— 微信小程序搜索“头豹”、手机扫上方二维码阅读研报

- ◆ 3、添加右侧头豹研究院分析师微信，邀您进入行研报告分享交流微信群



图说



表说



专家说



数说



扫一扫

实名认证行业专家身份

详情咨询



客服电话

400-072-5588



南京

杨先生： 13120628075

唐先生： 18014813521



上海

王先生： 13611634866

李女士： 13061967127



深圳

李先生： 18916233114

李女士： 18049912451

头豹 Project Navigator 领航者计划介绍



备注：活动解释权均归头豹所有，活动细则将根据实际情况作出调整。

头豹 Project Navigator 领航者计划与商业服务

- 头豹以**研报服务**为切入点，根据企业不同发展阶段的资本价值需求，以**传播服务、FA服务、资源对接、IPO服务、市值管理**为基础，提供适合的**商业管家服务解决方案**



扫描上方二维码
联系客服报名加入

备注：活动解释权均归头豹所有，活动细则将根据实际情况作出调整。

读完报告有问题？

快，问头豹！你的智能随身专家



扫描二维码
即刻联系你的智能随身专家

千元预算的
高效率轻咨询服务



STEP04 专业高效解答

书面反馈、分析师专访、
专家专访等多元化反馈方式



STEP02 云研究院后援

云研究院7×24待命
随时评估解答方案



STEP03 解答方案生成

大数据×定制调研
迅速生成解答方案



STEP01 智能拆解提问

人工智能NLP技术
精准拆解用户提问

