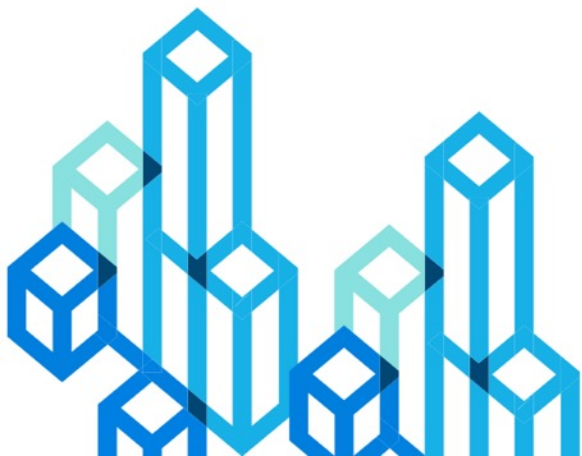


# 云原生时代的应用算力架构

@王晨纯



## 王晨纯(沐剑)

### 个人经历:

- 电商中台架构设计
- 阿里主站稳定性保障
- 异地多活/全链路压测等架构升级
- 聚石塔第一代PaaS容器平台/混合云方案
- 新零售品牌数字化/线上线下融合
- 政务服务数字化/云原生交付

### • 专业领域

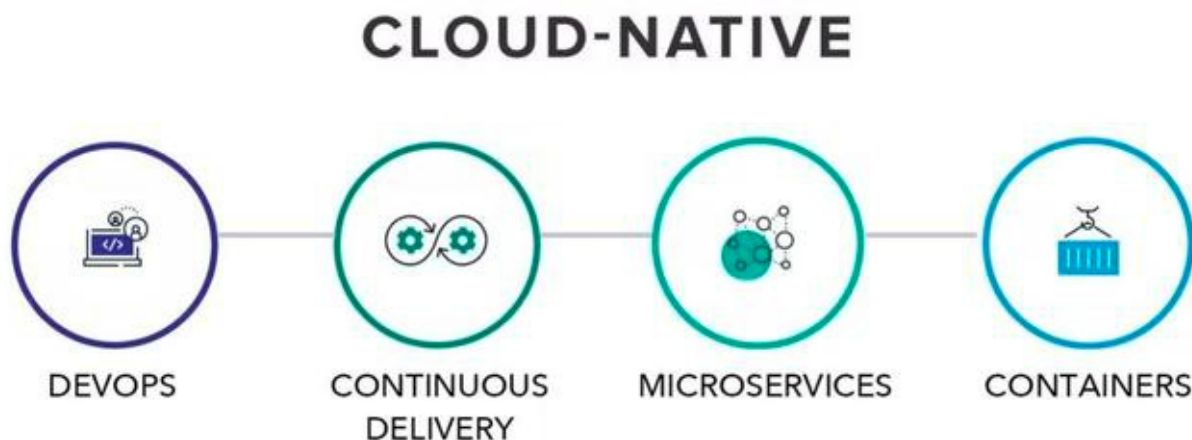
- 多年双11高可用核心保障成员
- 研发/运维/技术服务/中间件/架构/云原生



1. 如何理解云原生

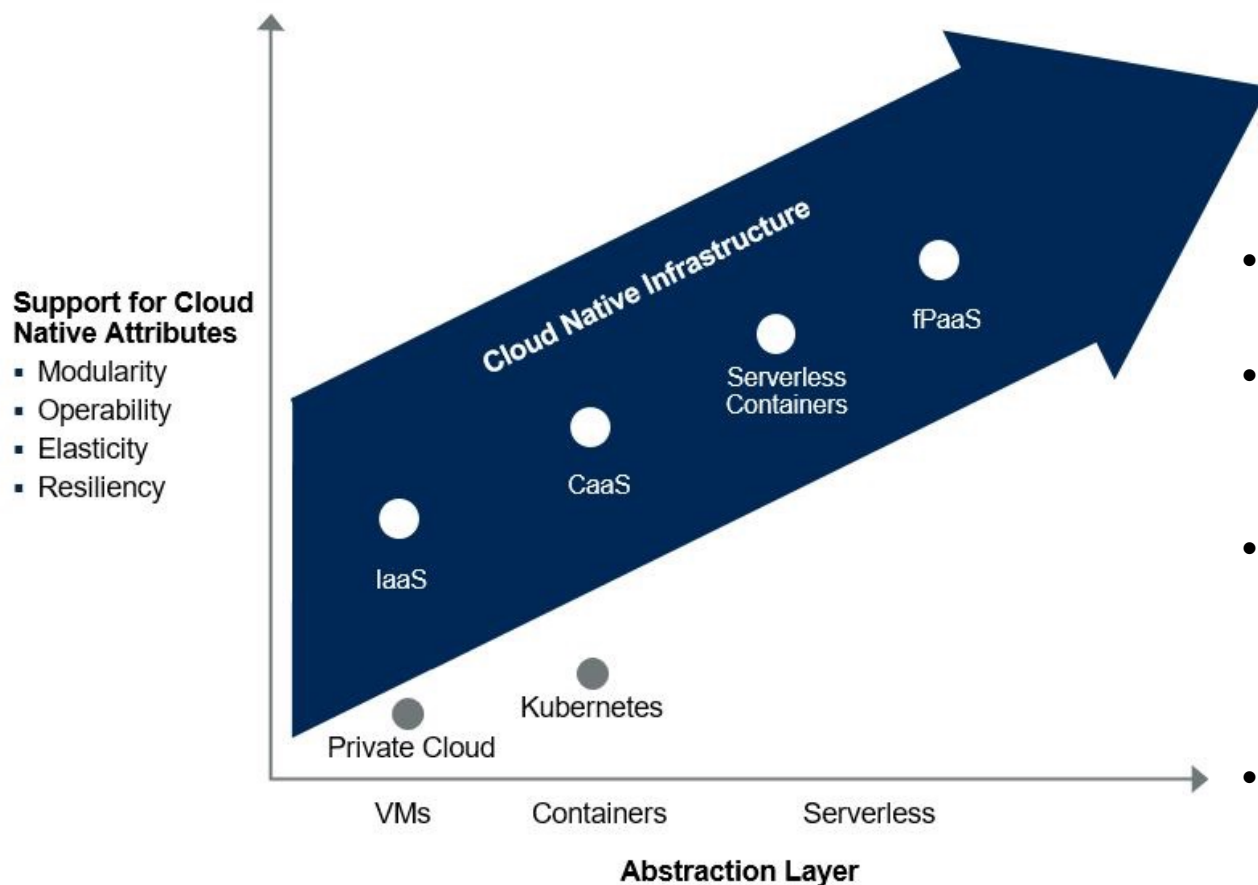
2. 上云后资源视角的转变

3. 从算力视角看应用架构设计



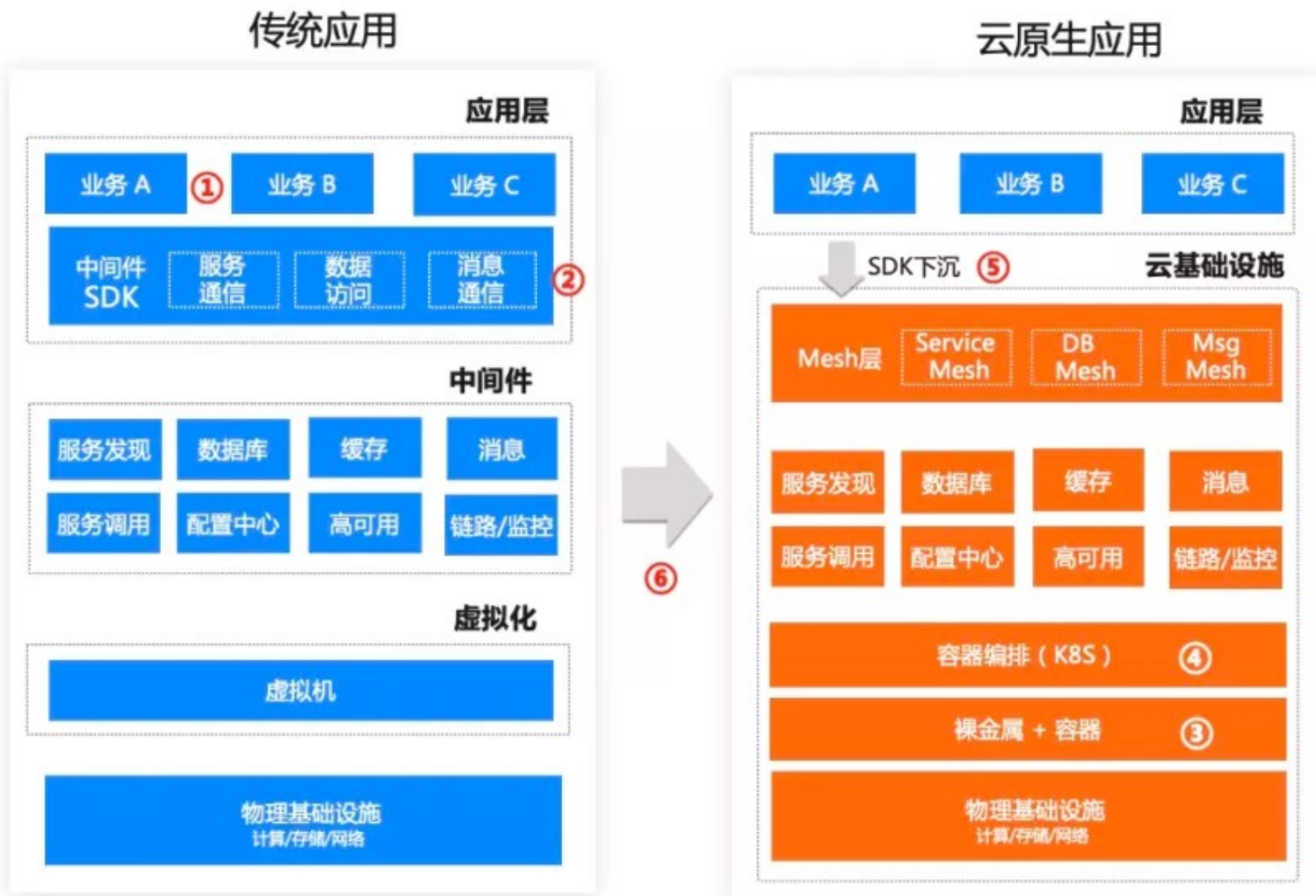
- 狭义的云原生和广义的云原生
- 微服务逐步融入云原生成为基础概念
  - 和SOA的本质区别
  - 服务拆分的粒度
- 让我们来谈一谈从算力视角去看今天的架构
  - 把「做什么」和「怎么做」解耦
  - Dapr / FaaS

# 1. 如何理解云原生



- 计算粒度越来越细
- 模块化程度UP
  - 从 package 到 image
- 资源调度自动化UP
  - 资源调度 / 弹性计算
  - 更关注业务逻辑
- 弹性效率UP
  - 物理机
  - VM
  - Container
  - Function
- 容灾效率UP

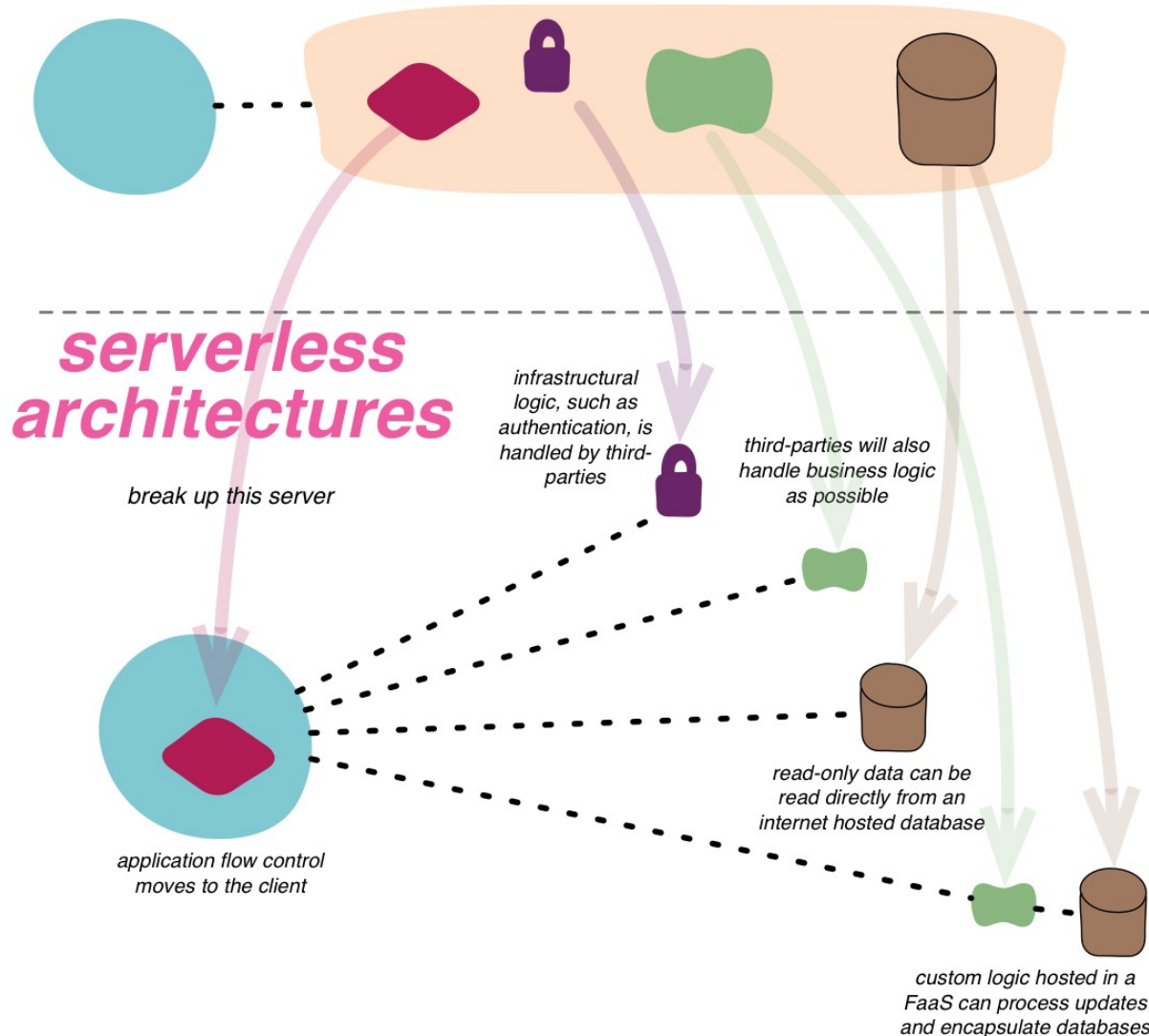
# 1. 如何理解云原生(e.g. 对Java部署架构的影响)



①模块化 ②池化弹性 ③自动化运维 ④容灾自愈

# 1. 如何理解云原生(热门的Serverless/FaaS)

A traditional internet delivered app has a client communicating with a long-lived server process that handles most aspects of the application's logic



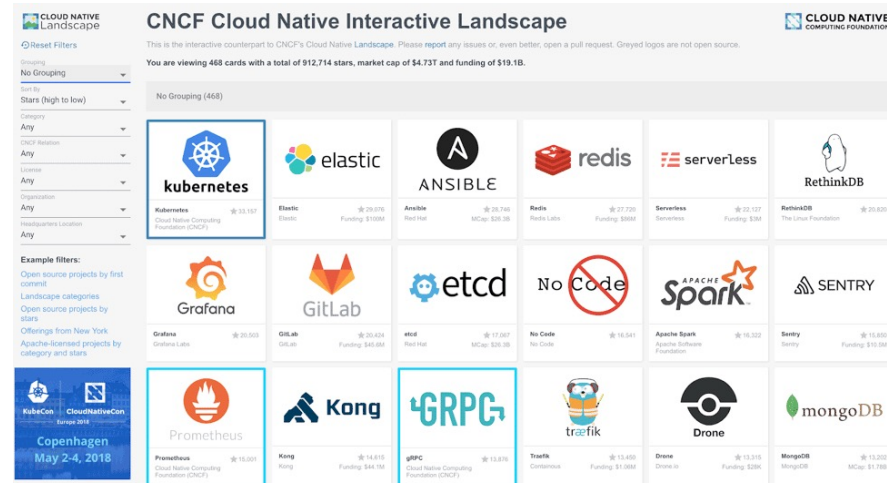
- 理性看待Serverless
  - 容器粒度差异
- 本质是算力的再分配
- 离线在线混部架构
- 离线在线异地架构
- FaaS
  - 适用场景
  - 把不同类型的算力用计算架构分离
  - 类比DBaaS



## 2. 上云后资源视角的转变



- 关注点上移(对资源描述方式的转变)
  - 物理资源 (服务器、硬盘、CPU)
  - 算力资源 (Core、IOPS、空间)
- 设计范式的转变
  - 怎么做
  - 做什么
- Vendor lock-in
  - 云原生时代的操作系统(K8S)
  - CNCF





### 3. 算力视角——从I/O看计算存储网络

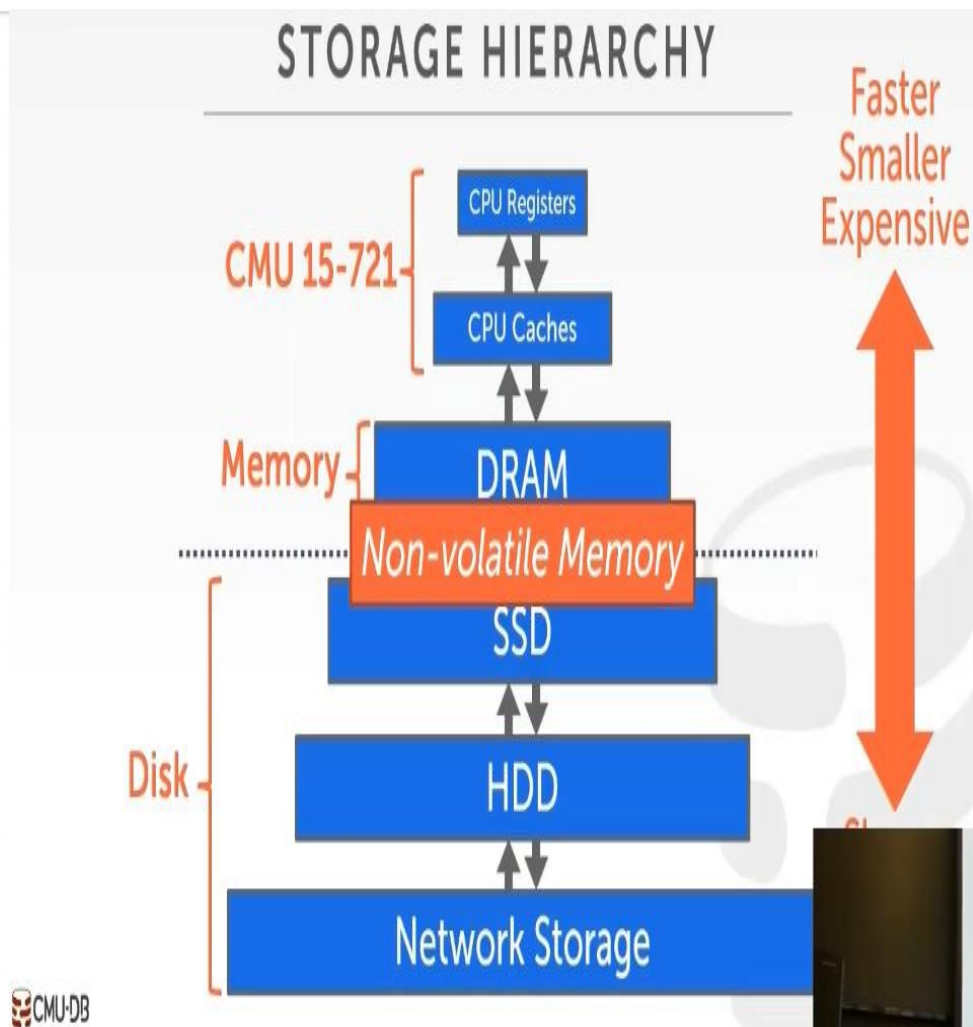


Srigi  
@srigi

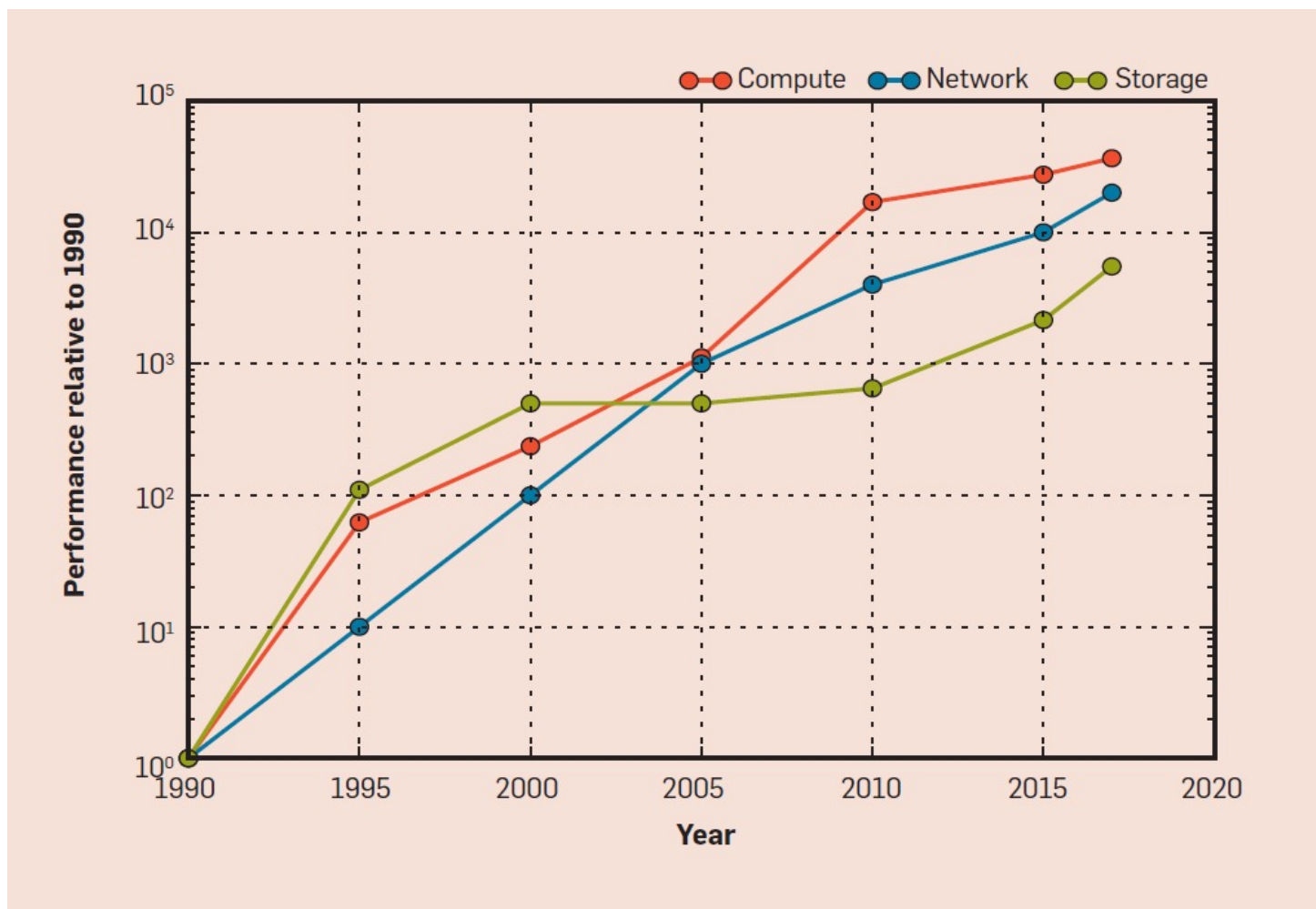
#### "Latency Numbers Every Programmer Should Know"

It is hard for humans to get the picture until you translate it to "human numbers":

1 CPU cycle	0.3 ns	1 s
Level 1 cache access	0.9 ns	3 s
Level 2 cache access	2.8 ns	9 s
Level 3 cache access	12.9 ns	43 s
Main memory access	120 ns	6 min
Solid-state disk I/O	50-150 $\mu$ s	2-6 days
Rotational disk I/O	1-10 ms	1-12 months
Internet: SF to NYC	40 ms	4 years
Internet: SF to UK	81 ms	8 years
Internet: SF to Australia	183 ms	19 years
OS virtualization reboot	4 s	423 years
SCSI command time-out	30 s	3000 years
Hardware virtualization reboot	40 s	4000 years
Physical system reboot	5 m	32 millenia



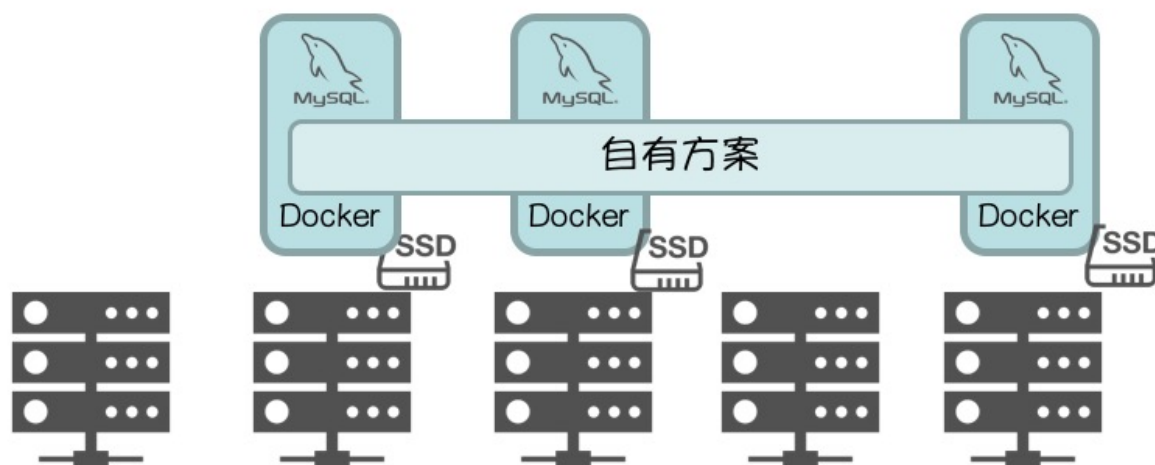
- 各种优化手段（拉近数据和计算）
  - 内存Cache
  - 冷热数据分离
  - 随机写改顺序写
  - MapReduce



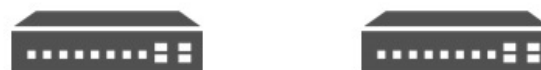
- 基础架构的升级带来设计逻辑的变化
  - HDD SSD RAM
  - 10G Network
- 让软件更好地利用硬件

计算存储分离从本质上让「状态」被解耦了  
计算和存储能力独立扩展

实例层



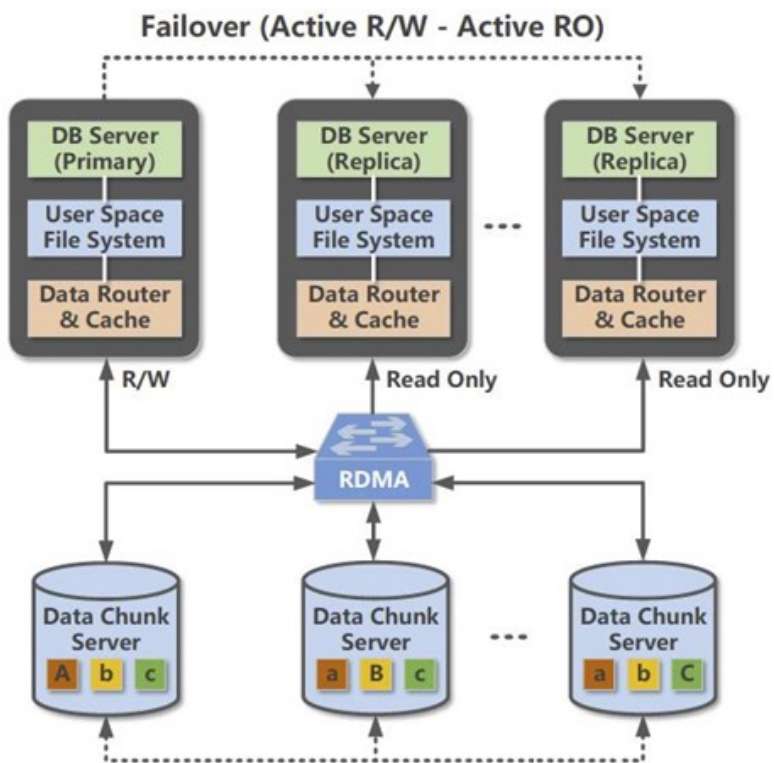
网络层



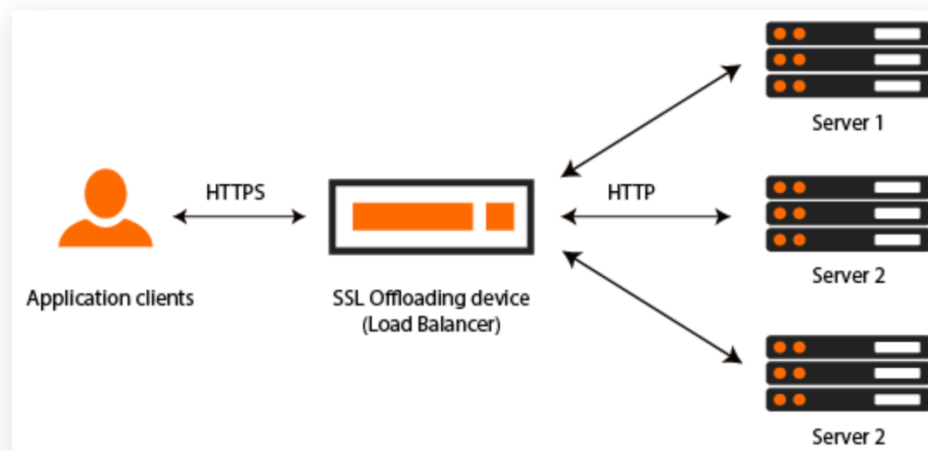
存储层



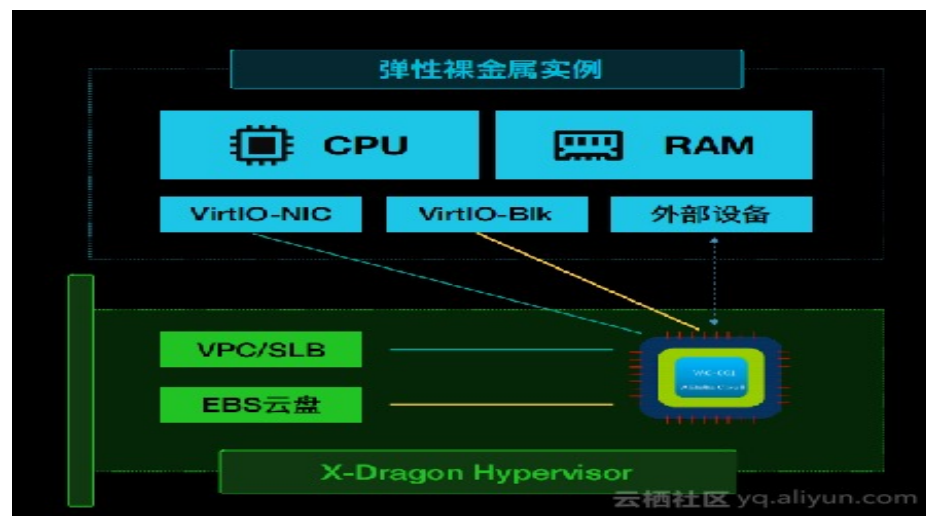
### 3. 算力视角——软硬一体架构(HPC的x86化)



DataBase based on RDMA

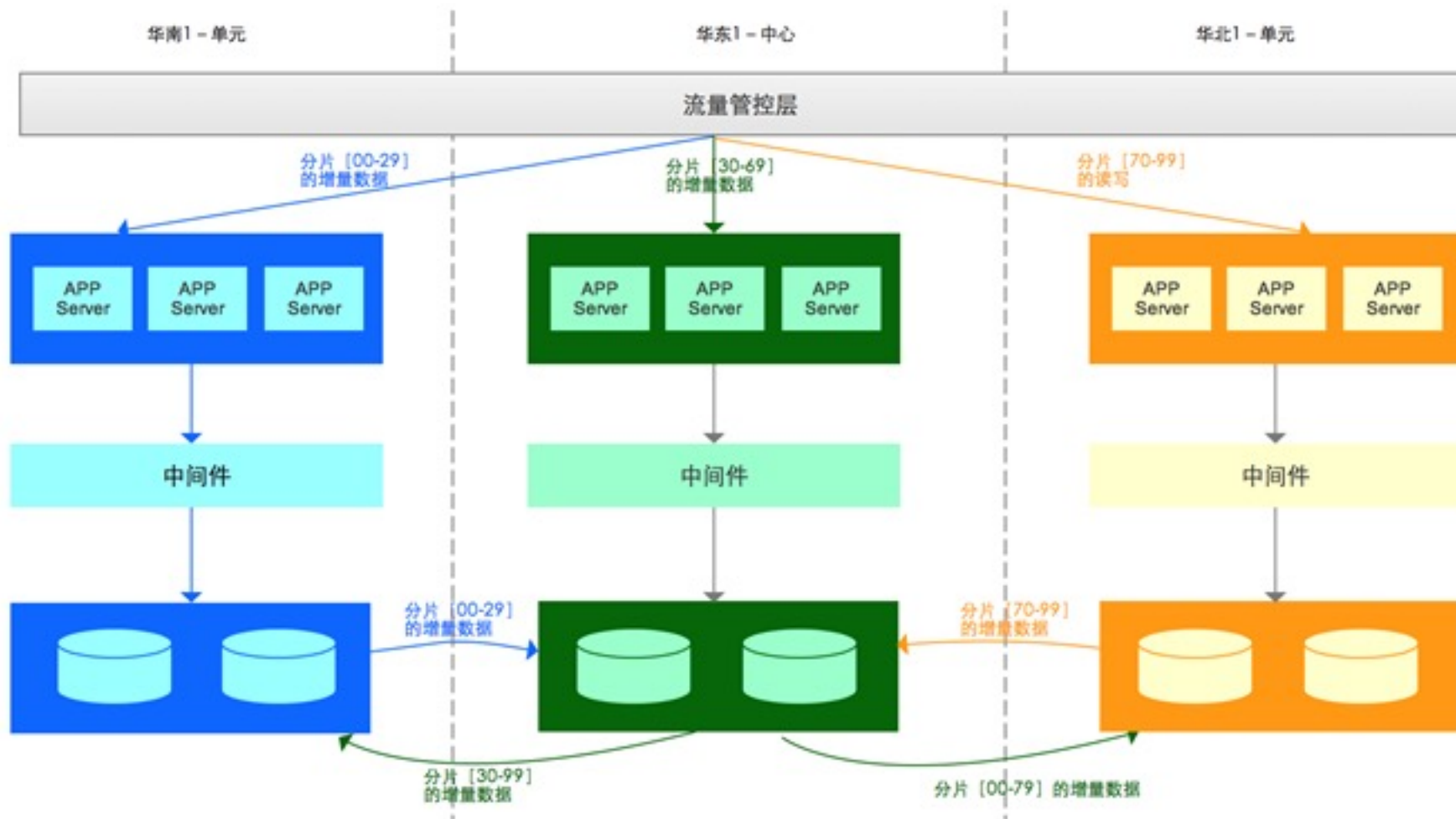


SSL Offload



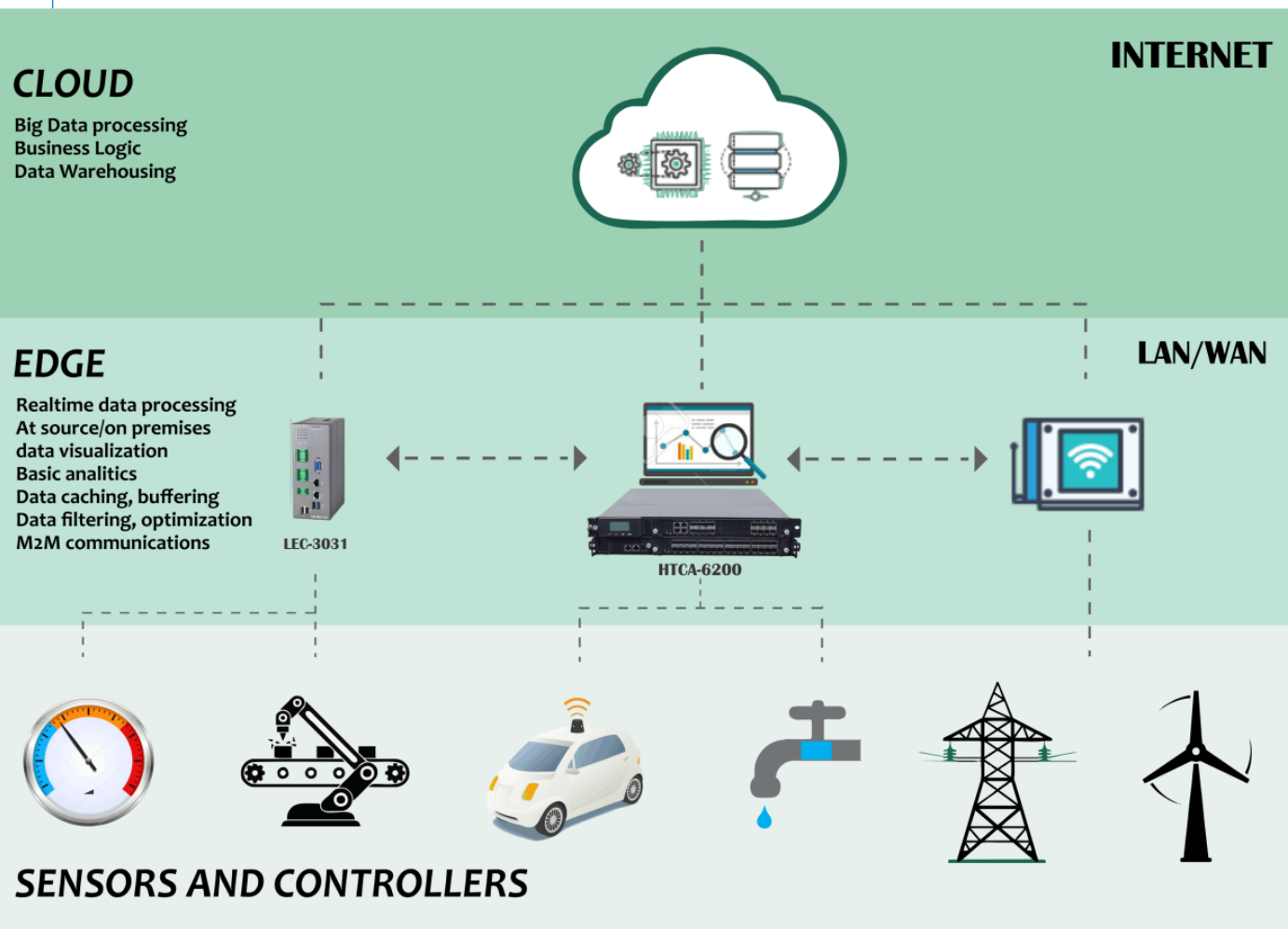
ElasticComputing Hypervisor Offload

### 3. 算力视角——异地部署



- 不同的目的导致不同的异地部署方案
  - 体验
  - 扩展性
  - 容灾
  - 单位计算成本

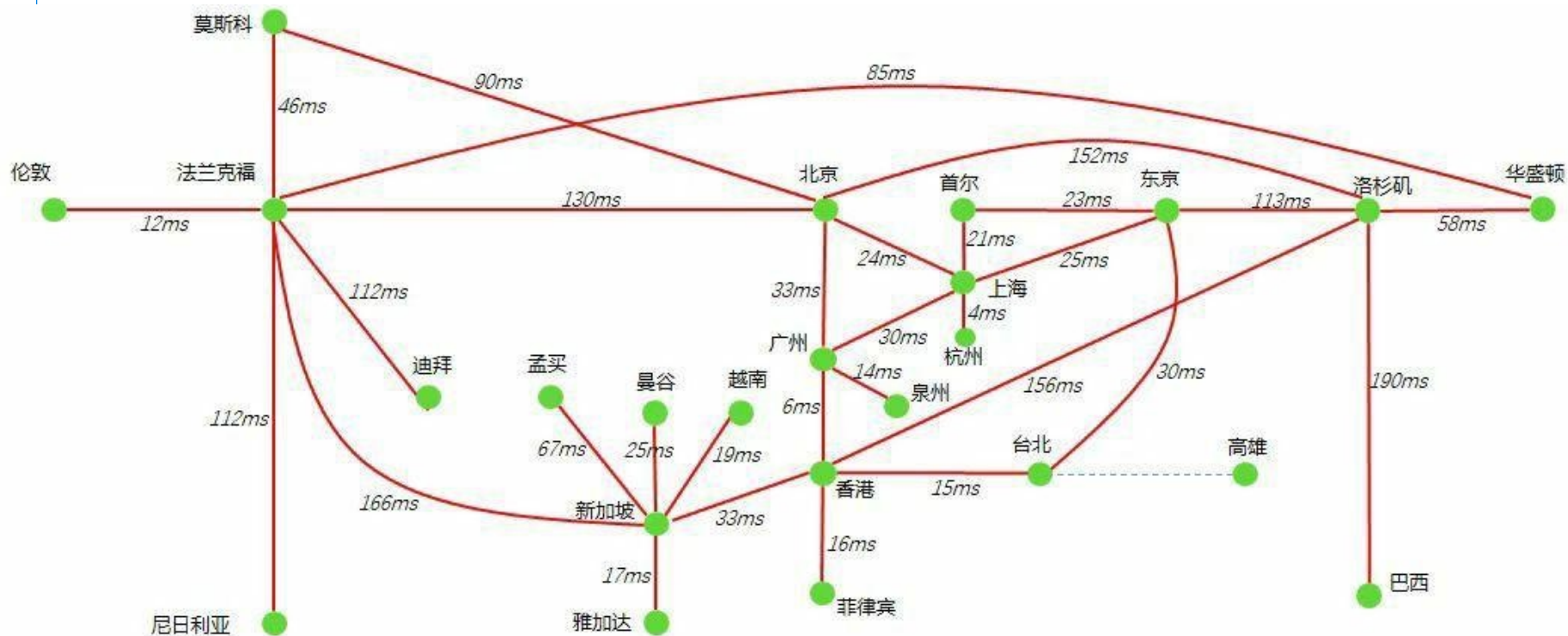
### 3. 算力视角——中心到边缘



- 让内容离用户更近
  - CDN
  - lua
- 让计算离用户更近
  - 游戏分区
  - 多Region部署
  - Edge
  - IoT
  - 工业互联网



### 3. 算力视角——全球部署



把地球看作一台计算机

- 就近接入
- 异地的容灾架构
- 数据一致性
- 最终一致性





麦思博(msup)有限公司是一家面向技术型企业的培训咨询机构，携手2000余位中外客座导师，服务于技术团队的能力提升、软件工程效能和产品创新迭代，超过3000余家企业续约学习，是科技领域占有率第1的客座导师品牌，msup以整合全球领先经验实践为己任，为中国产业快速发展提供智库。



高可用架构公众号主要关注互联网架构及高可用、可扩展及高性能领域的知识传播。订阅用户覆盖主流互联网及软件领域系统架构技术从业人员。高可用架构系列社群是一个社区组织，其精神是“分享+交流”，提倡社区的人人参与，同时从社区获得高质量的内容。