HCloud: 跨云的无服务计算平台

夏虞斌 · 上海交通大学 · 2019-11

JOINT CLOUD

IPADS
INSTITUTE OF PARALLEL AND DISTRIBUTED SYSTEMS

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

云际计算与无服务架构

云际计算：云计算发展的趋势

数据　　计算　　算法　　服务

云**际**计算是以云服务实体之间开放协作为基础，通过多方云资源(**包括云上数据资源**)深度融合，方便开发者通过**“软件定义”**方式定制云服务、创造云价值的新一代云计算模式

服务无边界　**云间有协作**　**资源易共享**　**价值可转换**

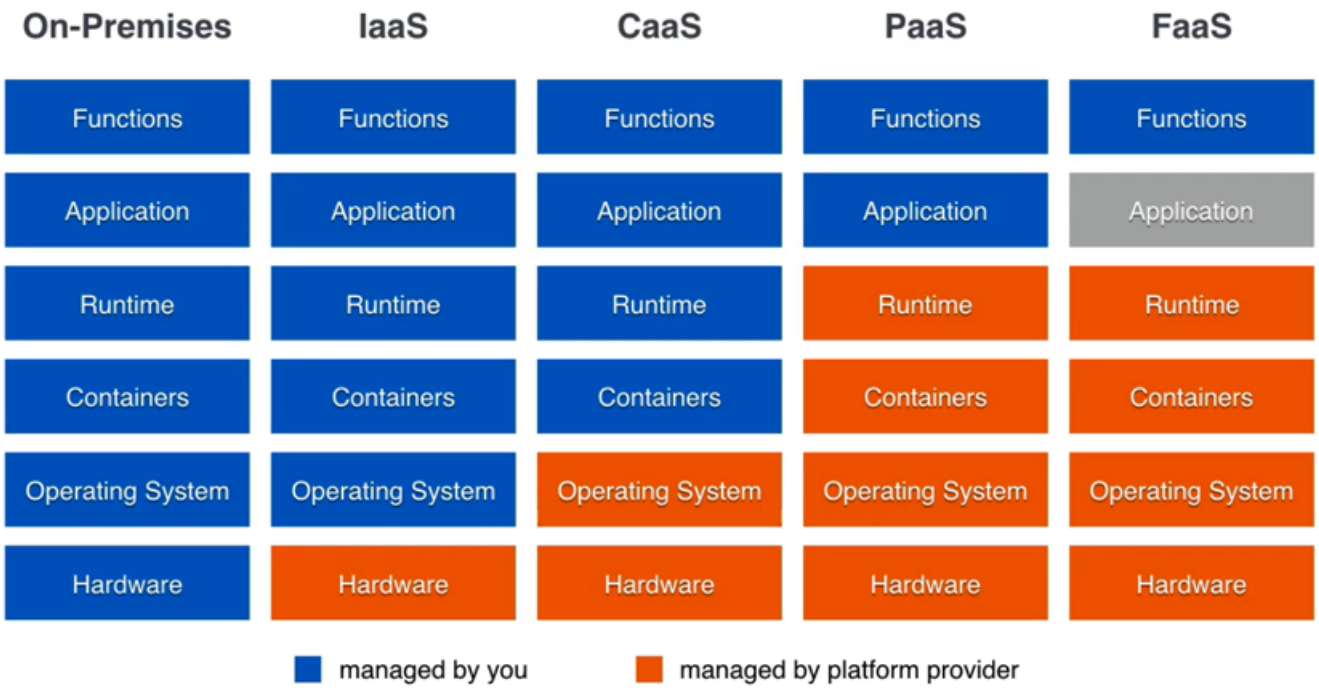
跨云计算的优势

- **价格优势**（以2VCPU/4G内存为例）
  - Amazon: 0.7/hr -> 0.13/hr（spot instance，无固定时间，提前2分钟通知）
  - 阿里云: 0.41/hr -> 0.06/hr （抢占式实例，固定1小时，提前5分钟通知）
  - 交大云: 闲置资源（非教学时间，负载周期相对稳定）
- **性能优势**
  - 更低的时延：分布式带来的 end-to-end 时延优势
  - 更强的扩展：由单一云资源扩展到多云资源
- **兼容优势**
  - 降低云供应商锁定问题的影响，允许用户在多云之间自由切换

跨云场景需要选择合适的粒度

- **方便实际部署**
  - 以传统的虚拟机/容器为粒度进行部署，跨云迁移的负载高
  - 部署过程需要多家云厂商协助，修改底层软件栈的成本高
- **控制性能损失**
  - 跨云平台相对单云平台可能引入额外的性能开销
  - 需要控制数据和计算由于跨云而产生的性能损失
- **保证计算安全**
  - 在多云与混合云计算环境下，信任分散导致责任分散

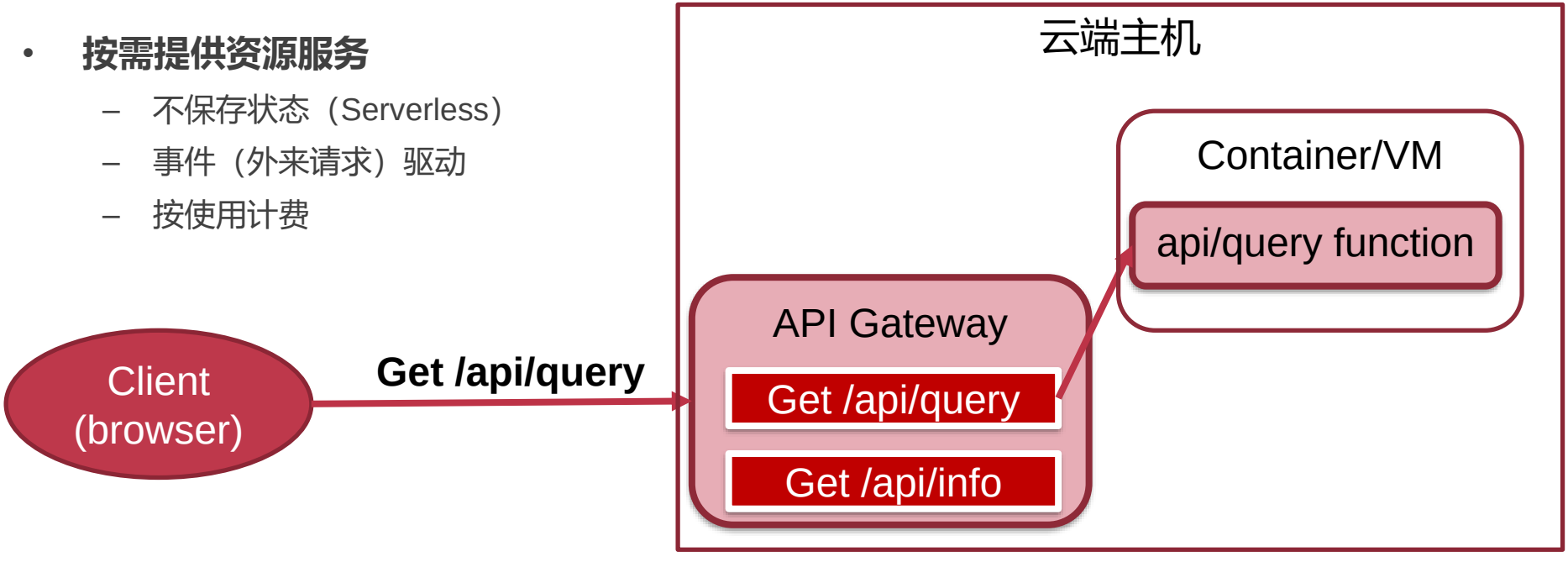
无服务架构：函数即服务（FaaS）



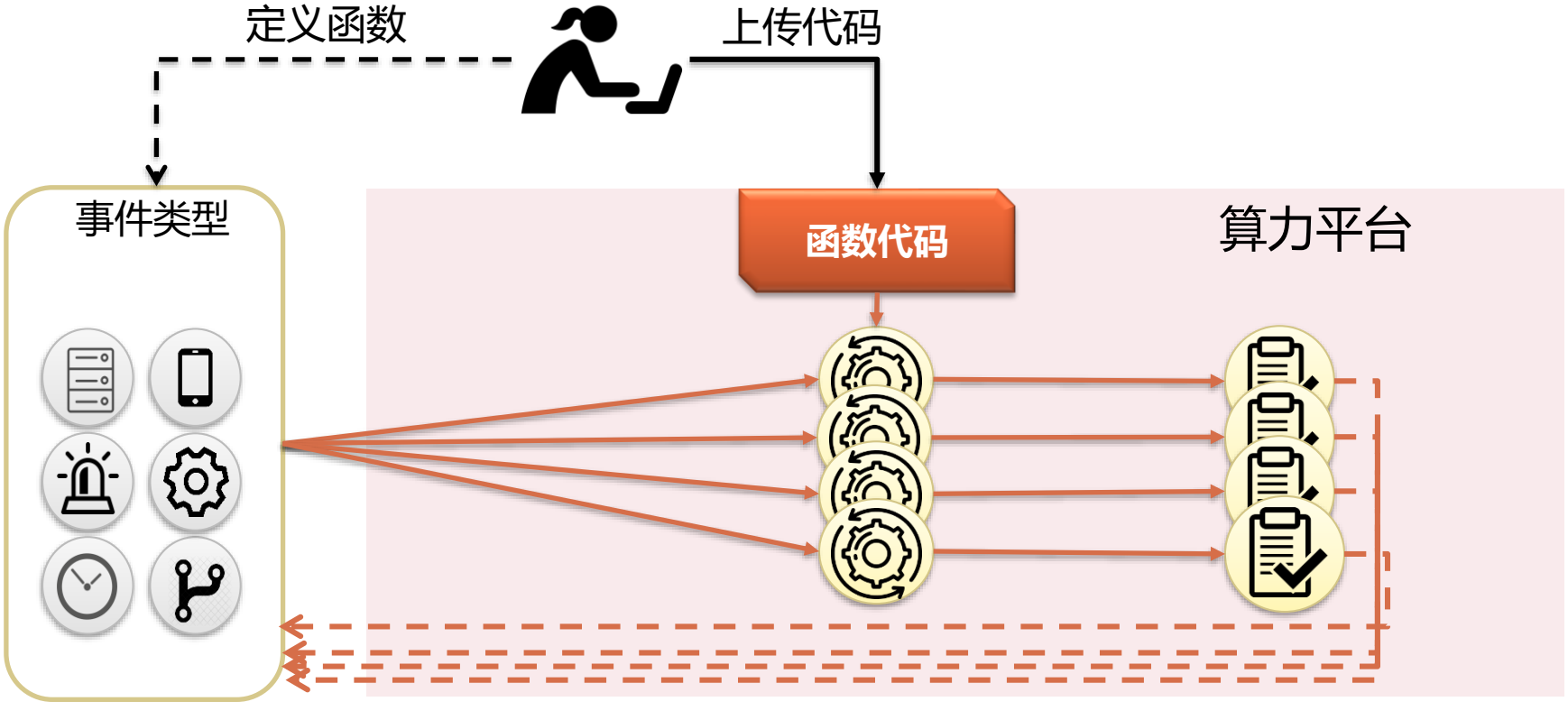
Amazon、微软、Google等均已推出无服务（Serverless）计算架构

无服务架构：函数即服务（FaaS）

- **函数即服务(FaaS)+(BaaS)**
- **按需提供资源服务**
  - 不保存状态（Serverless）
  - 事件（外来请求）驱动
  - 按使用计费

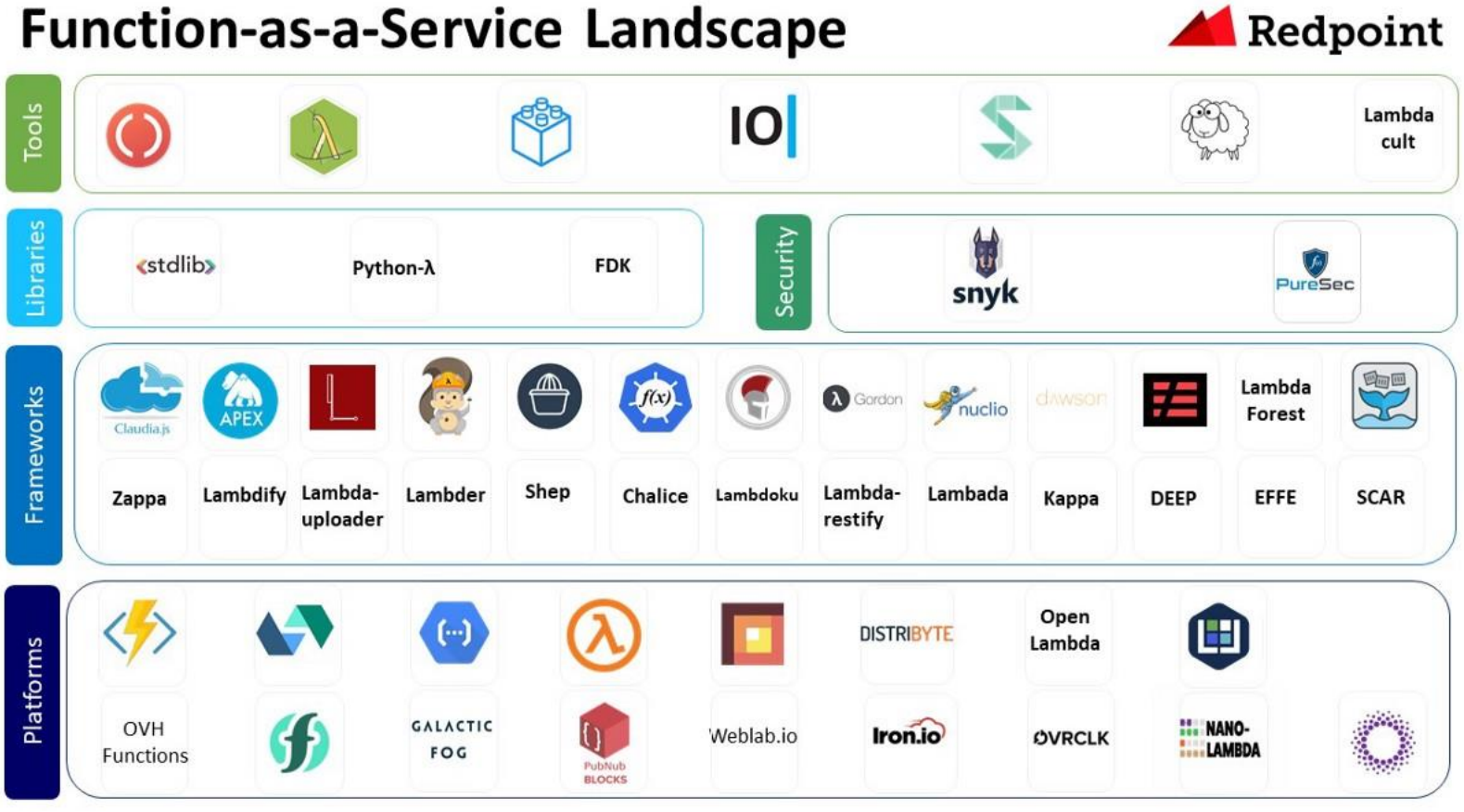


无服务架构：函数即服务（FaaS）



参考来源：SAND: Towards High-Performance Serverless Computing, ATC'18

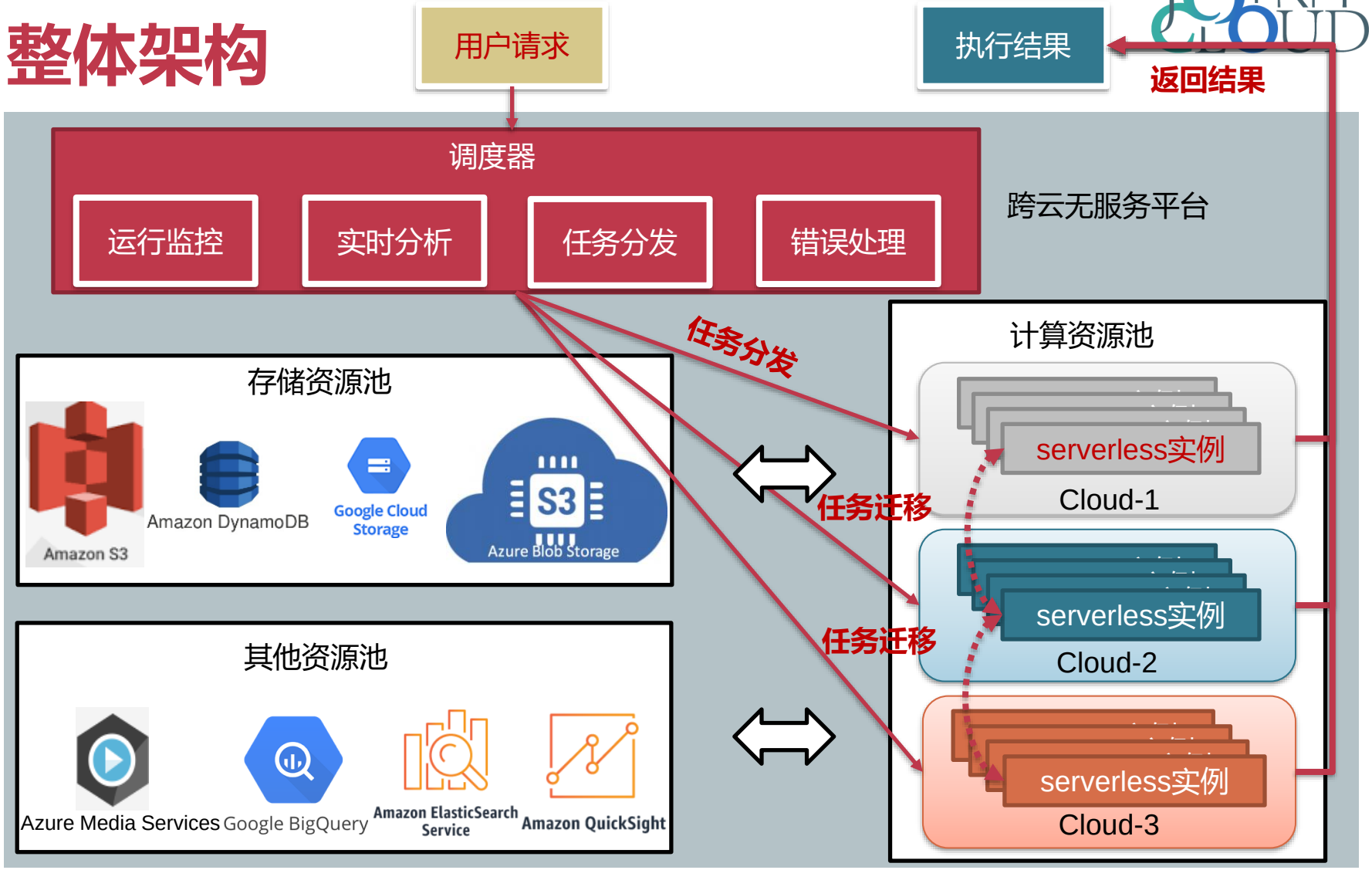
以FaaS为代表的云原生应用发展迅速



无服务架构在云际计算环境的优势

- **轻量部署**
  - 函数部署相对虚拟机和容器的成本更低，体现“无服务”的优势
- **高兼容性**
  - 部署过程对云透明，可基于传统的IaaS或容器，支持混合云
- **灵活计费**
  - 真正的按需计费，精确统计对各种资源的使用

整体架构


用户请求
执行结果
返回结果
调度器
运行监控
实时分析
任务分发
错误处理
跨云无服务平台
任务分发
计算资源池
存储资源池
Amazon S3
Amazon DynamoDB
Google Cloud Storage
S3
Azure Blob Storage
任务迁移
serverless实例
Cloud-1
serverless实例
Cloud-2
任务迁移
其他资源池
Azure Media Services
Google BigQuery
Amazon ElasticSearch Service
Amazon QuickSight
serverless实例
Cloud-3

跨云无服务架构：平台异构性挑战

- **计算资源的异构性**
  - 时间和空间层面闲置资源不同，可扩展能力不同
  - 计算硬件的多样性：安全处理器、AI加速器等
- **信任模型的异构性**
  - 不可信云平台对关键数据与用户隐私的保护
  - 消耗算力统计与收费模型的准确性

计算资源异构性的挑战

计算资源异构性挑战

	亚马逊AWS	微软Azure	谷歌Cloud Functions
Supported Languages	Node.js, Python, Java, C# and Go	C#, F#, Python, Java, Node.js, PHP	Node.js, Python
Persistent Storage	Completely stateless	Environment variables can be set, and can be stored into blob storage	Persistent storage available.
Max Code Size	50 MB compressed 250 MB uncompressed	None, you pay storage cost	100 MB compressed 500 MB uncompressed
Max Execution Time	900 secs.	600 secs.	540 secs.
Concurrent Functions	1K	depend on Triggers	1K
Billing factor	Execution time Allocated memory	Execution time Consumed memory	Execution time Allocated memory Allocated CPU

Source: RightScale

软件平台的异构性挑战

```java
package com.jlhood.retweetcounter.lambda;

import java.util.Map;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.APIGatewayProxyRequestEvent;
import com.amazonaws.services.lambda.runtime.events.APIGatewayProxyResponseEvent;

import com.google.common.collect.ImmutableMap;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.jlhood.retweetcounter.Leaderboard;
import com.jlhood.retweetcounter.dagger.AppComponent;
import com.jlhood.retweetcounter.dagger.DaggerAppComponent;

public class GetLeaderboardHandler implements RequestHandler<APIGatewayProxyReques
    private static final Gson GSON = new GsonBuilder().disableHtmlEscaping().creat
    private static final Map<String, String> CORS_HEADERS = ImmutableMap.of("Acces
```

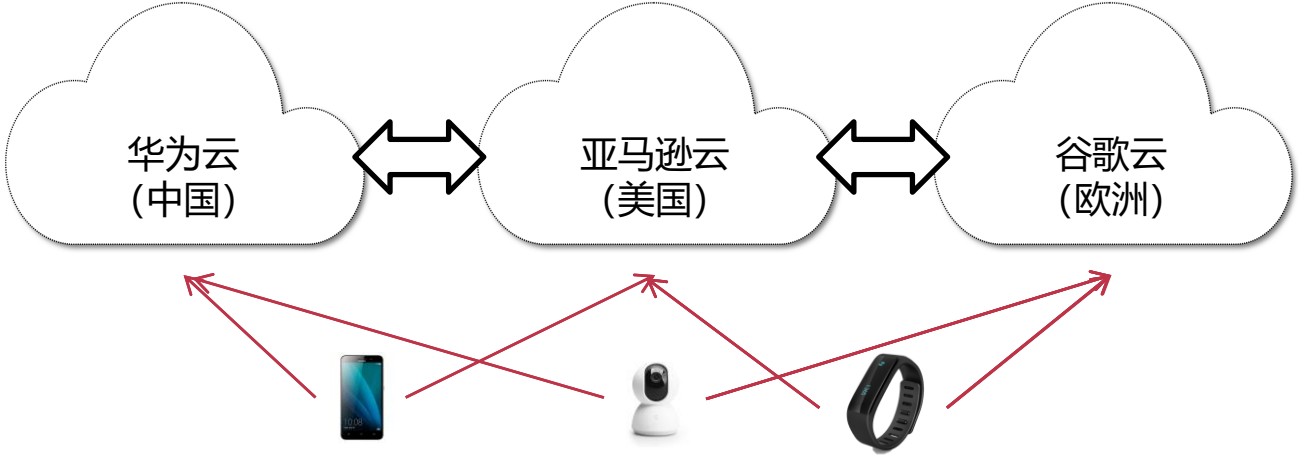
com.amazonaws.services.lambda

```node
var AWS = require('aws-sdk');
var doc = new AWS.DynamoDB.DocumentClient();

var config;

exports.handler = function(event, context) {
  if (config) {
    handleEvent(event, context);
  } else {
    var params = {
      TableName: 'MobileRefArchConfig',
      Key: { Environment: 'demo' }
    };
    doc.get(params, function(err, data) {
      if (err) {
        console.log(err, err.stack);
        context.fail(err);
      } else {
        config = data.Item;
        handleEvent(event, context);
      }
    });
  }
};

function handleEvent(event, context) {
  var cloudSearchDomain = new AWS.CloudSearchDom
```

source：https://github.com/aws-samples/

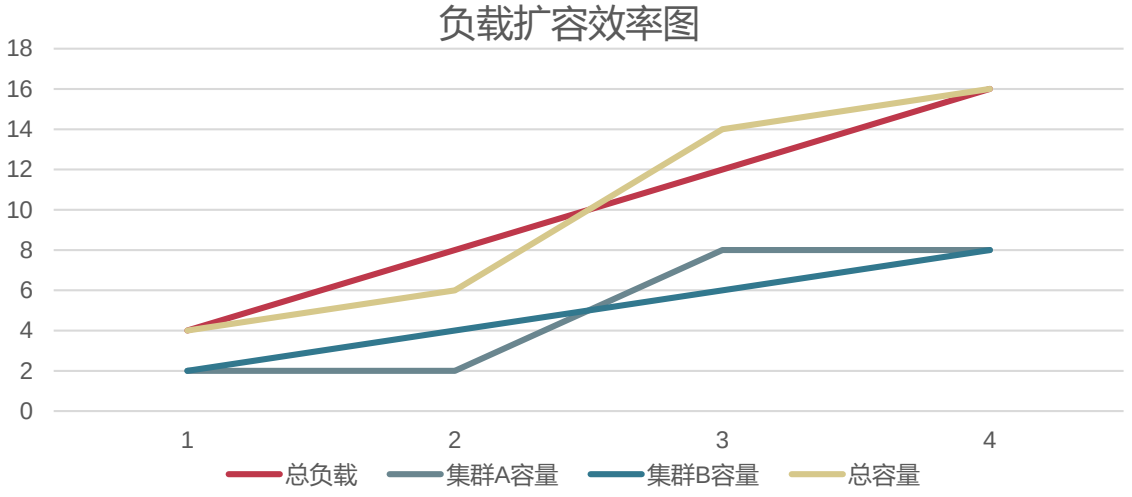
云际平台的动态扩容、弹性伸缩功能

- **自动切换功能：** 根据请求来源自动搜寻最近的服务器
- **自动资源伸缩：** 根据实时负载自动进行跨云资源的伸缩

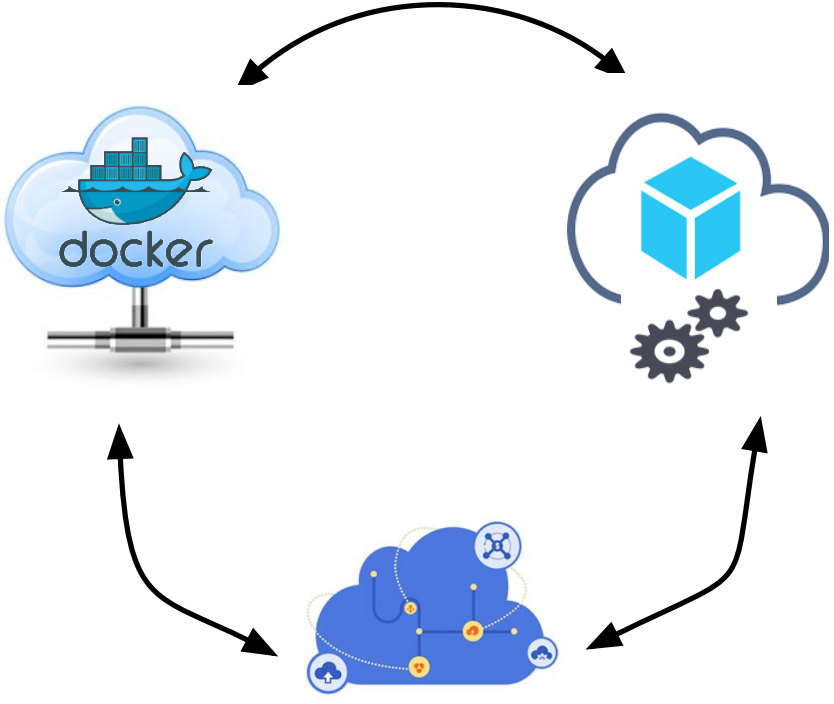


云际平台的动态扩容、弹性伸缩功能

- **异构的扩容速度**
  - 不同平台资源的启动速度不同：容器、虚拟机
  - 通过在扩容时的调度策略进行调整



高效的跨云计算迁移



迁移需求

云际复杂环境中数据倾斜等情况导致云际环境异常时延（拖后腿）问题

技术挑战

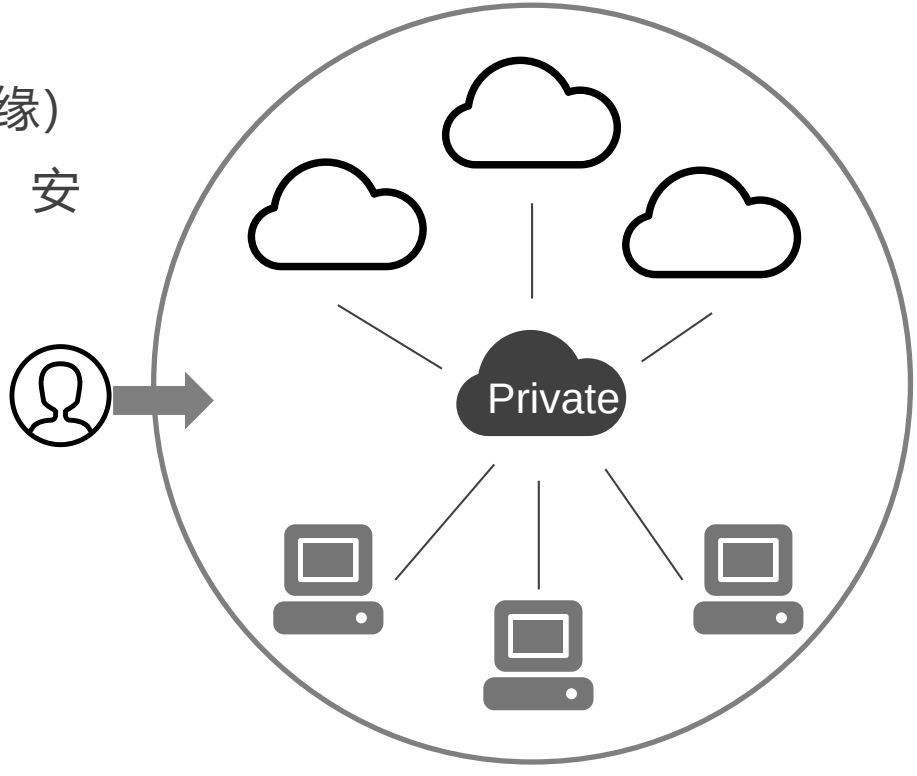
跨云平台造成的异构性（硬件异构性、虚拟化方法异构性等），将对跨云迁移造成挑战

蕴含问题

需要聚合跨云平台异构资源，研究云际应用与平台解耦的方法，实现高效安全的跨云迁移

高效的跨云计算迁移

- **跨云无服务框架**
  - 公有云、私有云以及其他节点（边缘）
  - 多种属性：延迟、SLA、异构硬件、安全性、价格、容量、带宽等因素
  - 提供标准抽象与统一接入
- **跨云无缝动态迁移**
  - 数据迁移与计算迁移的统一
  - 根据用户的位置等信息触发迁移
  - 迁移过程对最终用户透明



混合跨云无服务架构

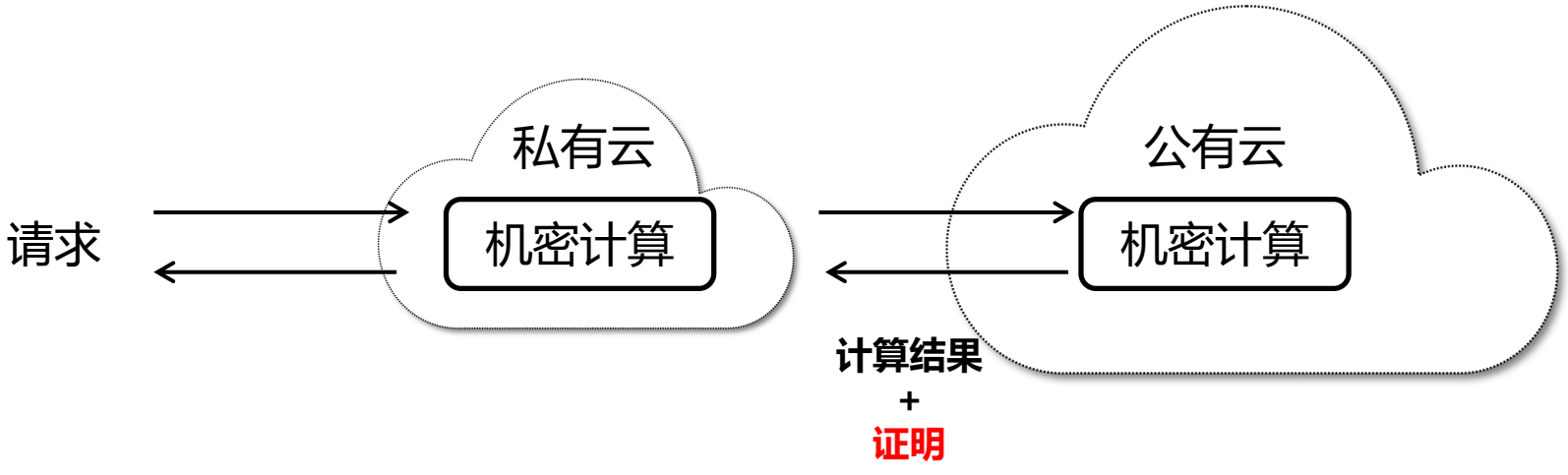
高效的跨云计算迁移

- **开发部署平台解耦**
  - 提高代码可迁移性
- **跨提供商部署**
  - 全平台调度，全自动横向扩展机制
  - 运行时动态迁移
- **最小化成本及最大化性能**
  - 全平台运行监测
  - 综合考虑安全、性能要求和成本的跨云调度决策

信任模型异构性的挑战

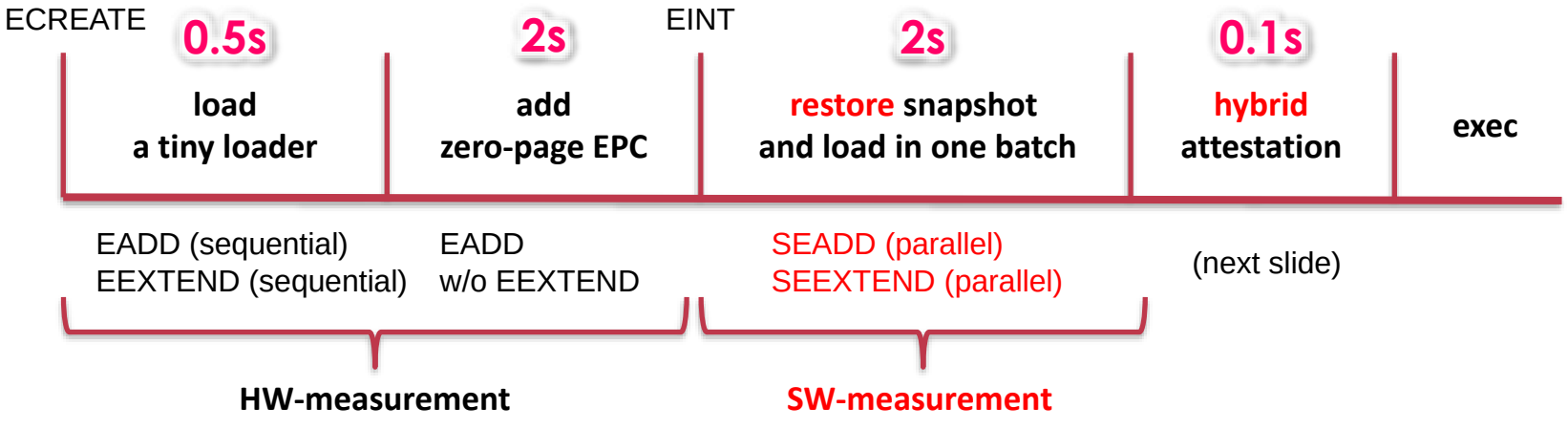
跨云场景的安全性挑战

- **分布式的信任问题**
  - 如何在打通多云平台的同时实现安全跨云计算
  - 如何保证数据在异构的多种云上不泄露
  - 如何为运算结果提供验证以防止恶意篡改



机密计算的优化

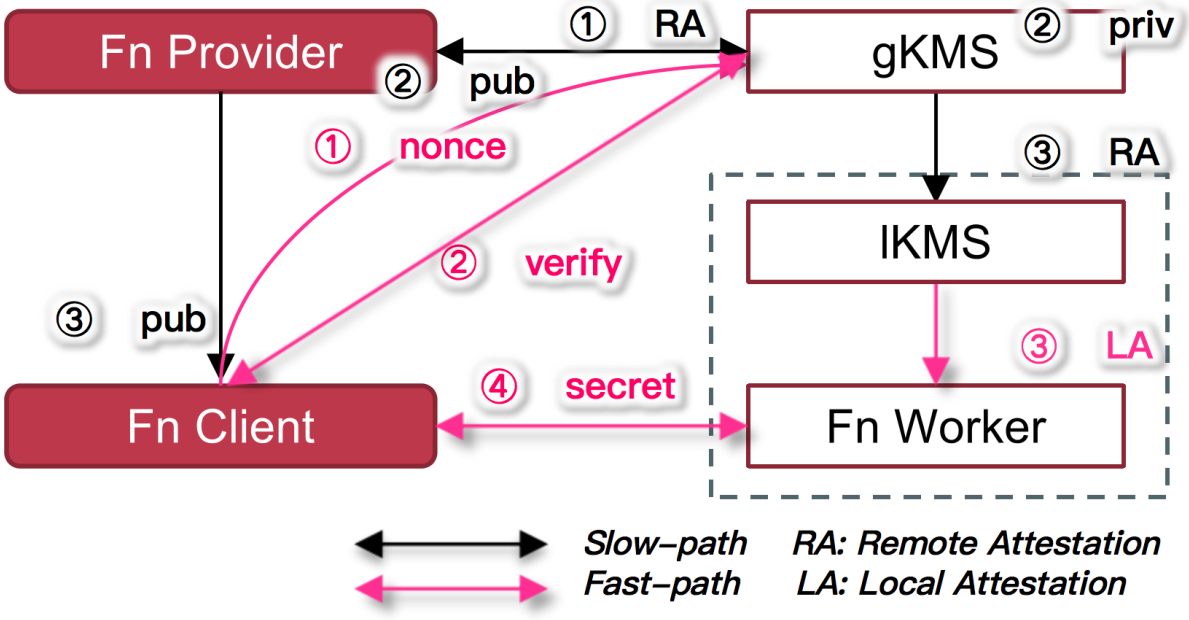
- Switch to SW-measurement



- SW-based allows for **scalable parallelism**
  - EADD (1.7k cycles) vs SEADD (1k cycles) => 2:1
  - EEXTEND (100k cycles) vs SEEXTEND (9k cycles) => 10:1

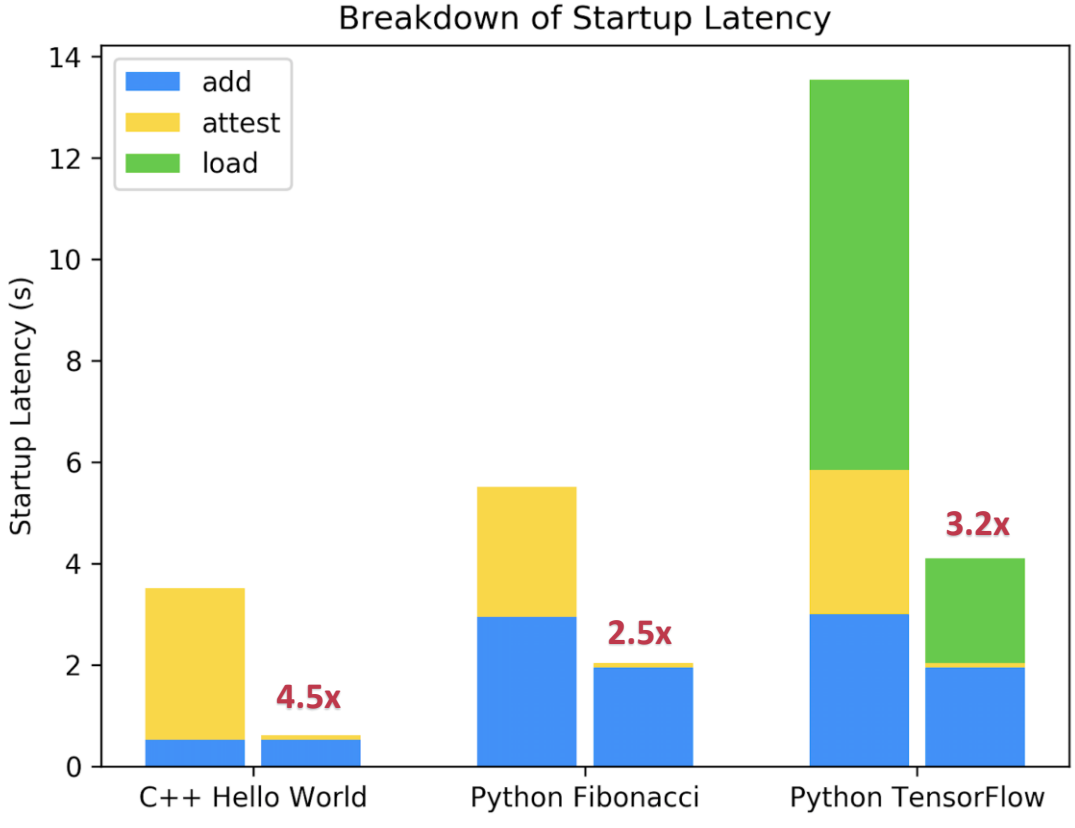
机密计算的优化

- Switch RA to Local Attestation (LA)



机密计算的优化效果

- Setup
  - Xeon E3
  - 4 Core
  - Disable HT
  - 128M PRM



跨云的资源使用计费挑战

- 如何实时追踪云收费信息，选择当前**性价比最高**的服务商
- 如何在计算环境不可信的前提下，对计算所消耗的资源进行**可信计费**

Azure Functions pricing

Azure Functions consumption plan is billed based on per-second resource consumption and executions. Consumption plan pricing includes a monthly free grant of 1 million requests and 400,000 GB-s of resource consumption per month per subscription in pay-as-you-go pricing across all function apps in that subscription. Azure Functions Premium plan provides enhanced performance and is billed on a per second basis based on the number of vCPU-s and GB-s your Premium Functions consume. Customers can also run Functions within their App Service plan at regular App Service plan rates.

METER	PRICE	FREE GRANT (PER MONTH)
Execution Time*	$0.000016/GB-s	400,000 GB-s
Total Executions*	$0.20 per million executions	1 million executions

AWS Lambda 定价

使用 AWS Lambda，您只需按使用量付费。我们将根据您函数的请求数量和持续时间

Lambda 定价

内存 (MB)	每个月的免费套餐秒数
128	3200000
192	2133333
256	1600000
320	1280000
384	1066667
448	914286

PoUW: Proof-of-Useful-Work

- **按实际运行的消耗计费**
  - 以运行代码数量（Basic Block）为粒度进行计费
  - 以运行有效工作的时间进行计费
  - 与无服务计算的原生计费模式一致
- **可信计费**
  - 基于可信硬件保证单次计费的安全性与可信性
  - 基于区块链等技术保证计费的不可篡改与可审计

总结

总结

- **云际计算与无服务架构的天然结合**
  - 轻量部署、高兼容性、灵活计费、面向云原生
- **无服务计算在云际环境下面临新的挑战**
  - 软件异构、硬件异构、性能异构、安全异构
- **面向云际的无服务架构设计**
  - 灵活利用资源，平衡性能成本，支持无缝迁移
  - 基于机密计算、区块链等技术，保障安全可控

下一步：打造跨云计算的基础平台

验证和奖励

元数据上链
e.g., 属性、价格、来源

需求上链
e.g., 出价、使用方式

数据拥有方

数据所有者可以是计算方，保证数据不出本地

请求访问

授权

算法仓库

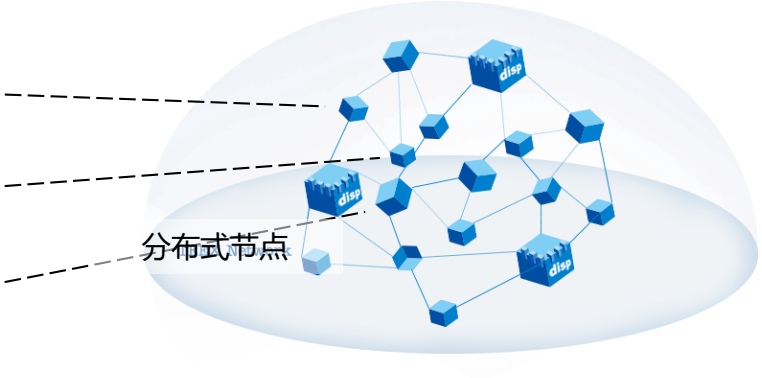
选择算法服务

数据使用方

数据

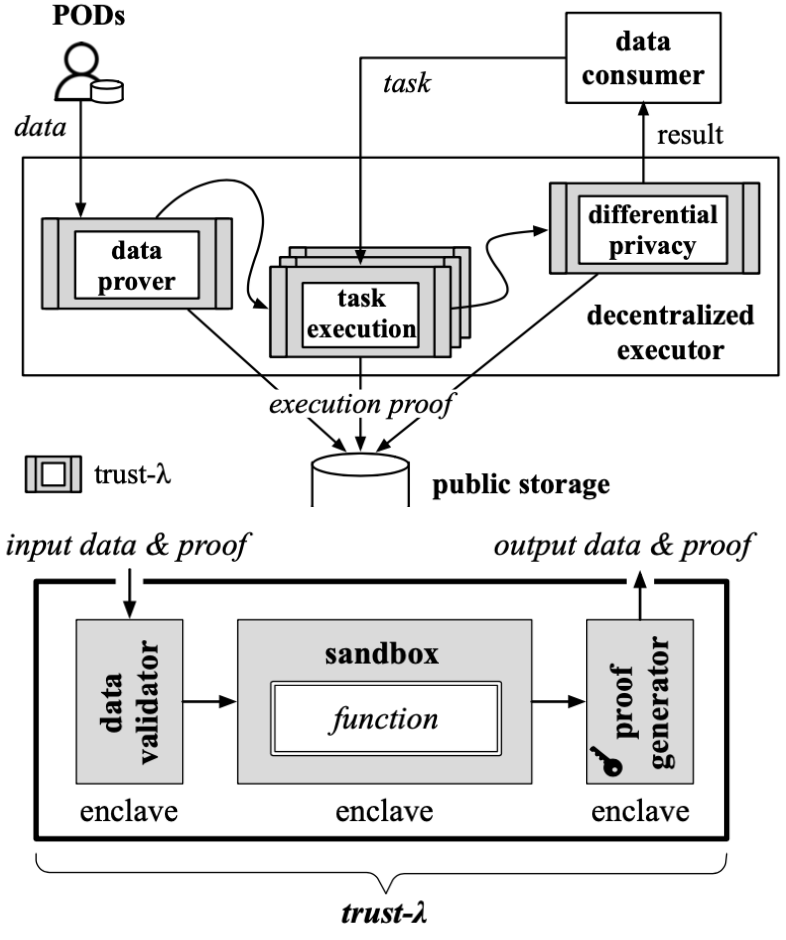
算法

算力提供方



基于貔貅OS的分布式机密计算云平台

- **数据的相对集中与隐私保护**
  - POD: Personal Online Data
  - by Tim Lee
  - 分布式的秘钥管理系统
- **计算的分散与可信增强**
  - 双向隔离，最小化可信基
  - 提出新的抽象：Trust-λ
  - 基于可信硬件的算力证明



谢谢！


IPADS
INSTITUTE OF PARALLEL
AND DISTRIBUTED SYSTEMS
上海交通大學
SHANGHAI JIAO TONG UNIVERSITY
交通大學