概念

> 云计算和区块链

云计算

基本概念

云计算：分布式计算、并行计算、网格计算

三种服务模式

- IaaS（基础设施即服务）：使用基础的运算资源。在按使用量付费的基础上，提供对基本计算资源的按需访问
- PaaS（平台即服务） ： 为软件开发者提供随需应变的平台-硬件、完整的软件栈、基础架构，甚至是开发工具，用于运行、开发和管理应用
- SaaS（软件即服务）：为用户提供的是一个完整的软件服务，可以通过 Web 浏览器、专用桌面客户端或者与桌面或移动操作系统集成的 API 来访问和使用该软件

云计算类型

- 公有云：用户可以通过互联网按需自助服务
- 私有云：企业根据自身需求在自家的数据中心上部署的专有服务，也可以托管在独立云供应商的基础结构上，或构建在位于异地数据中心的租用基础结构上（内部部署、外部托管）
- 社群云：由众多利益相仿的组织掌控及使用，例如特定安全要求、共同宗旨等。社群成员共同使用云端资料及应用程式。

- 混合云：公有云和私有云的结合
- 多云和混合多云：多云是使用来自两个或多个不同云提供商的两个或多个云；混合多云是将两个或更多公共云与私有云环境结合使用

涉及技术

数据存储

- 分布式存储、冗余存储
- 满足高吞吐率和传输率
- 目标：超大规模的数据存储、数据加密和安全性保证以及继续提高I/O速率
- 典型的分布式存储系统：Bigtable

资源管理与编程模型

- 除了数据存储，还要能面对海量数据进行分析处理
- 要求编程模型支持规模扩展
- 任务调度算法、任务容错

虚拟化技术

- IaaS的重要部分
- 特点：资源分享、资源定制（如CPU数目、内存容量、磁盘空间）、细粒度管理资源
- 虚拟机的快速部署、虚拟机在线迁移

虚拟化技术

引入VMM层，即虚拟机监控器，VMM管理分配硬件资源并管理虚拟机（VM）

- 传统架构是：硬件层 → OS → APP
- 虚拟化后：硬件层 → 资源管理层（VMM） → VM → OS → APP

X86架构指令级

- Ring0：内核态，权限最高，特权指令
- Ring1/2：X86下很少用
- Ring3：用户态，非特权指令，APP一般在这层

未虚拟化之前OS运行在内核态，对于运行在VM上的 GuestOS 会以为自己仍在 Ring0 然后下发特权指令，因此需要VMM对该特权指令拦截并处理，防止对硬件以及其他 GuestOS 造成影响，返回给该 GuestOS 的也是处理过的信息让其认为是在独占物理资源，即“特权——陷入模拟”

> 要注意有些非特权指令也有特权指令的效果，在X86架构设计之初，遗留了19条非特权指令，但是也有特权指令的效果，Guest OS下发此19条指令时也会造成对其他VM干扰

解决敏感指令（特权和非特权指令）三种方案

- 半虚拟化：将内核中的19条敏感指令改为不敏感
- 全虚拟化：无论特权/非特权全部拦截，但是会造成VMM资源消耗过大
- 硬件辅助虚拟化：支持硬件辅助虚拟化的硬件，可以自身筛选出来19条敏感指令，自动将19条敏感指令拦截并传递给VMM，由VMM处理之后，在返回给硬件。硬件辅助虚拟化的效率高，占用主机资源小

XEN架构下三种皆可，KVM架构下只能全虚拟化

虚拟化的类型

- 寄居虚拟化：在宿主OS之上，简单易实现，但占用宿主OS的资源开销较大
- 裸金属虚拟化：直接在硬件层上开发虚拟化层，难度大但是性能好，比如Xen
- OS虚拟化：在硬件层之上装入宿主OS，在宿主OS之上装容器，此时需要用到一个LXC的组件，而这个组件只有Linux有。因此容器里面的库，必须和Linux一样。比如 Docker
- 混合虚拟化：硬件层上装Linux，Linux内有一个虚拟化模块（KVM.KO）用于将内核虚拟化

虚拟化的特性

- 分区
- 隔离
- 封装
- 与硬件解耦

虚拟化架构大体上分类两种：XEN、KVM

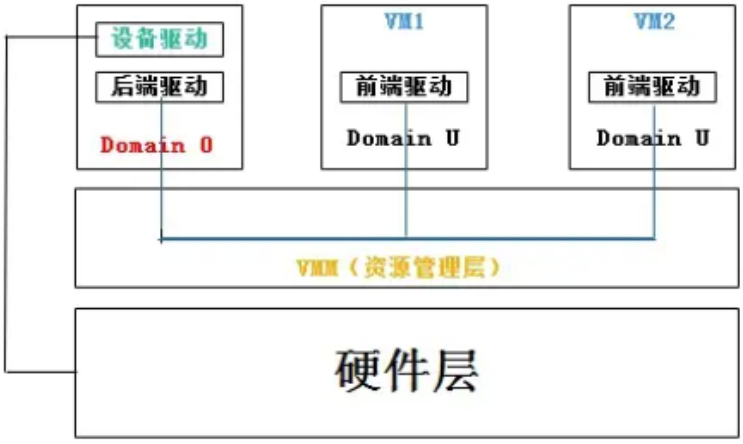
XEN

有一台运行在特权级别下的虚拟机（不是真正的VM），它有两个模块

- 后端驱动模块用于和 Domain 0 相连
- 设备驱动模块用于和硬件相连

VM通过前端驱动和 Domian 0 的后端驱动将IO刘发给 Domain 0 ，再由 Domain 0 的设备驱动完成对硬件的访问。缺点是IO路径长

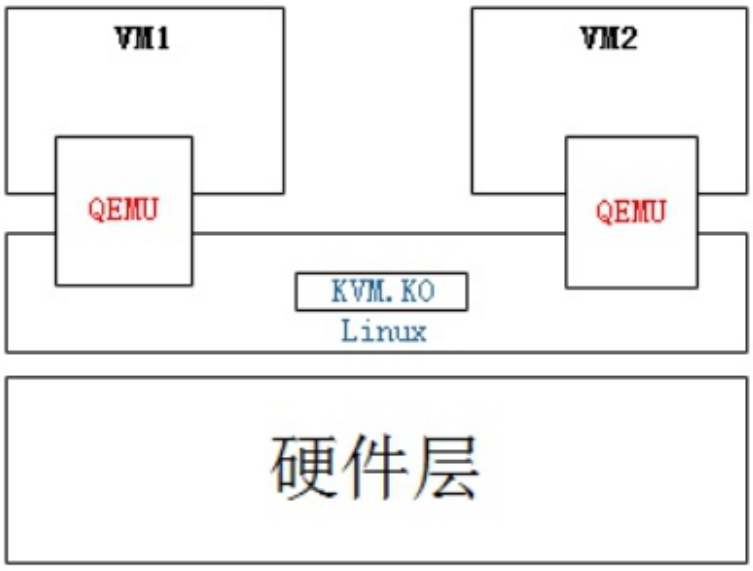
Xen架构：



KVM

在硬件层上有一个Linux，Linux中由一个KVM.KO虚拟化模块来将Linux内核转为VMM来提供虚拟化。也就是VM对于Linux来说就是文件，需要用QEMU（模拟仿真软件）来和Linux通信。缺点是不安全，没有VM控制台不能读VM监控

KVM架构：



对比

- KVM混合虚拟化，XEN裸金属虚拟化
- XEN中使用 Domain0 实现IO虚拟化，使用VMM实现VM之间的通信；KVM中用KVM.KO模块实现虚拟化，用QEMU实现通信
- XEN使用Domain0虚拟机，消耗更多底层性能，但是更安全；KVM性能好但通信不安全

分布式存储

分布式存储系统可在多个独立设备上分发数据。分布式网络存储系统采用可扩展的系统结构，使用多个存储服务器共享存储负载，利用位置服务器定位存储信息

云安全

安全体系建设两大部分

- 面向云平台侧：抗DDOS、下一代防火墙、IDS、APT攻击预警、防病毒、日志审计、数据库审计、运维审计、web业务安全审计、态势感知等安全设备等
- 面向用户侧：业务安全管理、用户管理、统一认证等层面来开展工作，需要做好入侵防御、访问控制、漏洞扫描、网页防算改、日志审计等

需求

- 按需分配
- 权限隔离
- 安全监测
- 安全防御
- 安全审计

IaaS：云平台的物理资源和虚拟资源的安全性
PaaS：在保证 IaaS 安全的基础上，保障 PaaS 平台自身的安全性
SaaS：保障 SaaS 平台自身安全和 SaaS 应用安全

区块链

基本概念

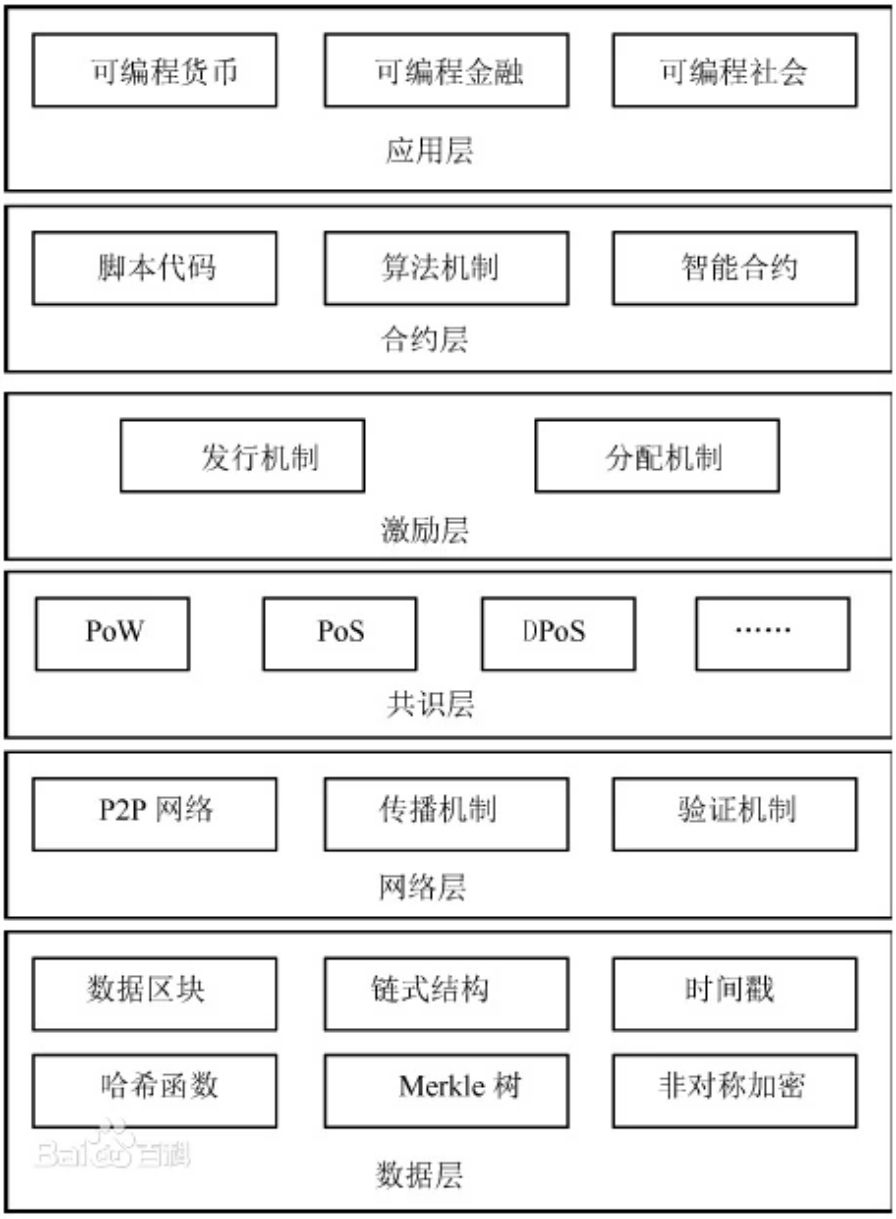
区块链本质上是一个去中心化的数据库

特点

- 去中心化
- 不可篡改：不受单个个体控制，在多台机子上复制存在，一般大于一半才能篡改
- 开放性：区块链技术基础是开源的，除了交易各方的私有信息被加密外，区块链的数据对所有人开放，任何人都可以通过公开的接口查询区块链数据和开发相关应用，因此整个系统信息高度透明
- 匿名性：各节点身份信息不需要公开或验证，信息传递可匿名；个人信息是加密的
- 独立性：基于协商一致的规范和协议（类似比特币采用的哈希算法等各种数学算法），整个区块链系统不依赖其他第三方，所有节点能够在系统内自动安全地验证、交换数据，不需要任何人为的干预

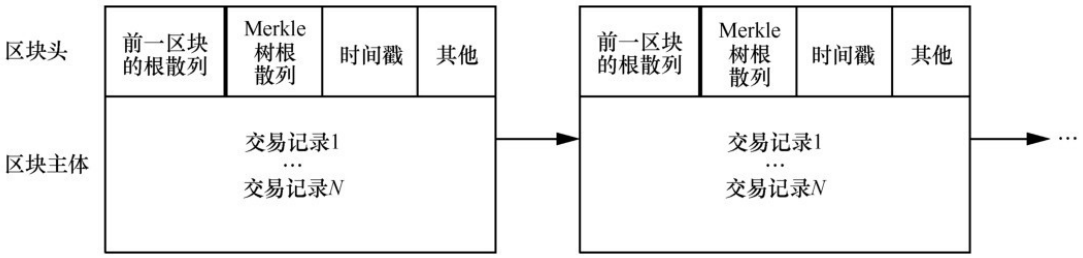
架构模型：

- 数据层：最底层，封装的链式结构、非对称加密、共识算法等实现数据存储和交易，块内可以采用 Merkle 树
- 网络层：对等节点组成网，采用P 2P
- 共识层：互不信任的节点通过某一机制在短时间内排除恶意节点的干扰,对正确结果达成一致,即称各节点之间达成共识，共识算法如 PoW、PoS
- （激励层）
- 合约层：智能合约，通过算法、程序编码等技术手段将传统合约内容编码成为一段可以在区块链上自动执行的程序
- 应用层组成



区块结构

- 区块头：记录当前区块的元信息
- 区块体：实际数据



工作

- 每个交易发生时都会被记录为一个数据区块，数据区块可以记录信息，如人、事件、时间、地点、数目、条件等
- 每个区块都与前后的区块连接，这些数据区块形成数据链，数据区块可以确认交易时间和顺序，将数据区块连接在一起可以防止任何数据区块被篡改或在两个数据区块之间插入其他数据区块
- 交易以区块形式组成不可逆的链即区块链，每添加一个区块都会增强对前一个区块的验证，也就增强了对整个区块的验证，增强不可篡改

区块链类型

- 公有区块链网络：任何人可加入，比如比特币
- 私有区块链网络：网络与公有的类似，但是有人管理网络控制批准谁准许参与网络，并执行共识协议、维护共享账本
- 许可式区块链网络：公有、私有区块链网络都可以成为许可式网络，会对获准参与的交易人员进行限制，虚获得邀请/许可才能进入
- 联盟区块链：多个组织分担维护区块链的责任，这些组织可以决定谁可以提交、访问数据

核心技术

分布式账本

- 交易记账由分布在不同地方的多个节点共同完成，而且每一个节点记录的是完整的账目，因此它们都可以参与监督交易合法性，同时也可以共同为其作证

非对称加密

- 区块链上交易信息公开，但账户身份是高度加密的，保障数据安全和个人隐私

共识机制

- 所有记账几点之间怎么达成共识去认定一个记录的有效性

- 少数服从多数、人人平等

智能合约

- 基于可信的不可篡改的数据，自动化执行一些预先定义好的规则，通过这些自动执行的规则，可以加快交易速度

共识算法

PoW

按劳分配，算力决定一切，谁的算力多谁记账的概率就越大。

比如比特币中使用的就是PoW，PoW计算一个数值（nonce），使得拼接上交易数据后内容的Hash值满足规定的上限。在节点成功找到满足的Hash值之后，会马上对全网进行广播打包区块，网络的节点收到广播打包区块，会立刻对其进行验证。最快的那个节点回将其添加到账本，其他节点复制

这个过程就是竞争记账，即如果想生成一个新的区块并写入区块链，必须解出比特币网络出的工作量证明谜题，谁先解出答案，谁就获得记账权利，然后开始记账并将将解出的答案和交易记录广播给其他节点进行验证，自己则开始下一轮挖矿

PoW工作量证明有三要素

- 工作量证明函数：比特币中使用 SHA256，一个哈希算法
- 区块：
  - 区块头：父区块哈希、版本、时间戳、难度、随机数、子区块、Merkle 根
  - 区块体：是 Merkle 树
- 难度值：
  - 难度值随网络变动，在比特币中，比如
    新难度值 = 旧难度值×（过去2016个区块花费时长/20160分钟）；
  - 还有一个目标值：目标值 = 最大目标值/难度值 ，最大目标值是固定的，若过去2016个区块花费时长少于20160分，那么这个系数会小，目标值将会被调大些，反之，目标值会被调小，因此，比特币的难度和出块速度将成反比例适当调整出块速度
  - 区块头中有一个教 MerkleRoot 的 hash 值，是通过 Merkle 树算出来的，作为交易聊表的摘要存在区块头中

也就是说，区块头中，只有 nonce 外其余全是明确的，解题的核心就在于调整 nonce 值对区块头进行计算，比哦SHA256

单个节点的工作量证明计算流程

- 生成 Merkle 根哈希，即节点自己生成一笔交易，与其他即将打包的交易通过算法一起生成的
- 组装区块头：是一个输入参数，由 Merkle 根哈希和其他区块头一起组成的
- 计算输出：工作量证明的输出 $= SHA256(SHA256(\text{区块头}))$ ，如果小于目标值就完成，反直再执行该函数，递归至小于目标值

共识记账流程

- 客户端产生交易后先全网广播
- 每个节点收到请求后将交易纳入区块，执行上面的工作量证明
- 当某个节点找到了证明，也就是成功解题，向全网广播
- 只有该区块的交易有效且之前未保存时，其他节点才认同该区块的有效性
- 接受该区块并在该区块末尾制造新的区块

优缺点

- 优点
  - 完全去中心化，任何人可加入
  - 节点自由进出，易实现
  - 破坏系统需要花费很大成本：需要掌握51%的算力；且在指定时间内给定一个难度，找到答案的概率唯一地由所有参与者能够迭代哈希的速度决定。与之前的历史无关，与数据无关，只跟算力有关
- 缺点
  - 对节点的性能网络环境要求高
  - 浪费资源
  - 效率低，每秒可能只能做几笔交易
  - 不能确保最终一致性（为啥？）

PoS

股权证明，不需要 PoW 中证明记账前的工作，需要证明拥有的某些财产，谁股权大谁记账概论就大

工作机制，以 Peer Coin 为例

- 币龄，币的数量×天数；币天。持有货币的时间
- 竞争开始前以自己的币龄下注，成为记账验证者
- 随机选出一个记账者
- 被选中的币龄清零，获得利息=币龄×5%/365，每被清空的这么多币获得的利息

也就是说，每个加入的都是持币人，成为验证者，根据持币多少选择谁有权利生产区块，以区块链中最长的链为准

优缺点

- 优点
  - 性能高，不需要无用计算
  - 人人可获得利息/挖矿，不会因算力集中导致中心化
  - 即使有51%的算力也未必能进行51%的攻击，因为有部分货币不是挖矿产生的而是由利息产生的，这就要求攻击者还需持有51%的货币量
- 缺点
  - 币无法发行：一旦开始挖矿，只有创世区块中有币，则其他矿机没法参与
  - 需要给币设计一个时间上限，因为一旦囤积很久的币，很久后发起攻击，币龄将极高

DPoS

代理权益证明，节点选举若干代理人，由代理人验证和记账，牺牲部分去中心化来得到性能提升

工作机制

- 节点根据所持有的加密货币数量占总数的百分比来投票，不是一人一票
- 投票选出代表 BP ，即可信节点，这些可信节点之间完全对等；要是算力不稳定等会被撤销
- 由可信节点进行决策，根据比例分

优缺点

- 优点
  - 记账节点减少，交易速度快
  - 一般不会发生链分叉并不可逆
  - 解决PoW资源消耗问题
- 缺点
  - 通过选举 BP 来出块记账，但是牺牲了中心化

智能合约

智能合约是存储在区块链上的程序，在满足预先确定的条件时会运行这些程序。 智能合约通常用于自动执行协议，以便所有参与者都可以立即确定结果，而无需任何中间人参与，也不会浪费时间。 智能合约还可以自动完成工作流程，在满足条件时触发下一个操作

遵循简单的“if/when…then…” 语句，这些语句被写入区块链上的代码中。当满足并验证预先确定的条件时将执行操作

三要素

- 自治：合约一旦开启就会自动运行，不需要人为干预
- 自足：通过提供服务或发行资产来获取资金，也需要使用这些资金（？）
- 去中心化：不依赖单个中心化服务器，而是分布式的，通过网络节点自动运行

一份合约的构建需要：多方用户共同参与制定一份智能合约 → 合约通过P2P网络扩散并存入区块链 → 自动执行

区块链安全

区块链技术生成的数据结构本身具有安全质量。它基于密码学、去中心化和共识原则，可确保对交易的信任。
在大多数区块链或分布式账本技术（DLT）中，数据被结构化为块，每个块都包含一笔交易或一组交易。每个新块都以加密链的形式与之前的所有块相连，几乎不可能被篡改。区块内的所有交易都通过共识机制进行验证和商定，确保每笔交易的真实性和正确性

攻击手段

- 网络钓鱼获取用户凭证
- 路由攻击：区块链依赖于大数据传输，在数据传输到互联网中可以拦截数据，切区块链参与者通常看不到该威胁
- 女巫攻击：创建并使用许多虚假网络身份来淹没网络使系统崩溃
- 51%攻击：超过50%的算力就意味着可以控制账本并操纵它。私有区块链不易受到51%攻击

区块链安全

- 身份和访问管理
- 密钥管理
- 数据隐私
- 安全通信
- 智能合约安全
- 交易背书