

# Substrate 区块链应用开发

测试和上线

---

孙凯超

kaichao@parity.io

获取帮助: <https://substrate.io>

# 内容

---

- Benchmark 确定交易权重
- 切换 PoA 为 PoS
- Runtime 的配置参数
- Chain Spec 和部署公开测试网络
- 安全审计和通证经济模型
- 总结和展望

# Benchmark 确定交易权重

---

总费用 = 基本费用 + (字节费用 + 权重费用) \* (1 + 动态调节费率) + 小费

权重被用来定义交易产生的计算复杂度：

- 合理的权重值需要通过 **benchmark** 来获取
- 可调用函数的注释中要给出计算复杂度和数据读写操作
- 通过WeightToFee, 转换权重值为权重费用

# Benchmark 硬件配置

- CPU, Intel i7-7700K - 4 c / 8 t @ 4.20GHz
- Memory: 64GB, DDR4, 2400 MHz
- OS: Ubuntu 20.04
- Rust version: rustc 1.43.0 (4fb7144ed 2020-04-20)

## Rise-2 New configuration

**Processor:** Intel i7-7700K - 4 c / 8 t - 4.2 GHz / 4.5 GHz

**Memory:** From 32GB

**Storage:** NVMe, SATA available

**Public bandwidth:** 500 Mbps

**Private bandwidth:** -

From  
**£56.53**  
ex. VAT / month

[Configure](#)

Available in 6 datacentres

Delivery from 120s



# Benchmark 可用选项

---

## Runtime 执行方式:

- Native
- Interpreted Wasm (wasmi)
- Compiled Wasm (wasm-time)

## Database 选择:

- RocksDB (default)
- ParityDB
- SubDB

# Benchmark 数据库读写

---

Database read & write:

// 1  $\mu$ s  $\rightarrow$  1\_000\_000 weight

```
pub const DbWeight: RuntimeDbWeight = RuntimeDbWeight {  
    read: 25_000_000, // ~25  $\mu$ s  
    write: 100_000_000, // ~100  $\mu$ s  
};
```

# Benchmark 可调用函数

---

Demo 代码:

<https://github.com/kaichaosun/play-substrate/pull/1/>

文档:

<https://www.shawntabrizi.com/substrate-graph-benchmarks/docs/#/>

# Benchmark 可调用函数

---

编译和运行：

```
cargo build --features runtime-benchmarks --release
```

```
./target/release/node-template benchmark \  
  --chain dev \  
  --execution=wasm \  
  --wasm-execution=compiled \  
  --pallet benchmark-demo \  
  --extrinsic do_something \  
  --steps 20 \  
  --repeat 50
```



# 切换 PoA 为 PoS

---


## 切换方法：

- [Substrate node](#),
  - 删除不需要的模块
  - 添加自己的模块
- node template,
  - Aura -> Babe
  - 添加 staking 相关模块
  - 添加治理模块

# 切换 PoA 为 PoS

---

切换方法：

- [Substrate node](#),
  - 删除不需要的模块
  - 添加自己的模块
-  node template,
  - Aura -> Babe
  - 添加 staking 相关模块
  - 添加治理模块

# 切换 PoA 为 PoS: BABE

---

BABE 区块生成算法的特点：

- 出块节点随机
- 同时存在次级出块节点
- 当长时间不出块，会导致网络瘫痪

对应[代码](#)。

# 切换 PoA 为 PoS: Staking

---

关联的模块:

- staking, session
- authorship
- offences, grandpa, im-online
- utility

对应的[代码提交记录](#)。

# 切换 PoA 为 PoS: 治理

---

关联的模块:

- treasury
- collective
- membership, elections-phragmen
- democracy, scheduler

对应的[代码](#)。

# Runtime 配置参数

---

常见的配置项为：

- Runtime 常用的类型别名如 BlockNumber, Balance ...
- 区块生成时间
- WeightToFee
- 初始区块配置
- .....

检查每个模块所用的配置是不是符合业务需求。

# Chain Specification

---

Chain Spec 文件包含：

- 元信息如 name, id, chainType
- 启动节点 bootNodes
- telemetryEndpoints
- protocolId
- properties (tokenSymbol, tokenDecimals)
- genesis 信息

[Chain Spec 例子](#)

# Chain Specification

---

如何生成 Chain Spec :

- 修改 chain\_spec.rs, command.rs
- 添加初始区块账户: `subkey generate`
- 添加验证人账户和 Session Keys
  - for i in 1 2 3 4; do for j in stash controller; `do subkey inspect "$SECRET//$i//$j"; done; done`
  - for i in 1 2 3 4; do for j in babe; `do subkey --sr25519 inspect "$SECRET//$i//$j"; done; done`
  - for i in 1 2 3 4; do for j in grandpa; `do subkey --ed25519 inspect "$SECRET//$i//$j"; done; done`



# Chain Specification

---

如何生成 Chain Spec :

代码示例

- 修改 chain\_spec.rs, command.rs
- 添加初始区块账户: `subkey generate`
- 添加验证人账户和 Session Keys
- 生成 Chain Spec

```
./target/release/node-template build-spec --chain tao-staging > tao-staging.json
```

- 编码 Chain Spec

```
./target/release/node-template build-spec --chain=tao-staging.json --raw >  
tao-staging-raw.json
```

# Chain Specification

---

启动 bootnode:

```
./target/release/node-template \
```

```
--node-key c12b6d18942f5ee8528c8e2baf4e147b5c5c18710926ea492d09cbd9f6c9f82a \
```

```
--base-path /tmp/bootnode1 \
```

```
--chain tao-staging-raw.json \
```

```
--name bootnode1
```

# Chain Specification

---

启动验证人：

```
./target/release/node-template \  
  --base-path /tmp/validator1 \  
  --chain tao-staging-raw.json \  
  --bootnodes  
/ip4/your-ip/tcp/30333/p2p/12D3KooWBmAwcd4PJNJvfV89HwE48nwkRmAgo8Vy3uQEYNNH  
Box2 \  
  --name validator1 \  
  --validator
```

也可将 bootnode 的配置信息添加到Chain Spec中。

# Chain Specification

---

验证人启动之后，添加 Chain Spec 配置的对应验证节点的 BABE和 GRANDPA 使用的key,

```
curl http://localhost:9933 -H "Content-Type:application/json;charset=utf-8" -d "@babe1"
{
  "jsonrpc":"2.0",
  "id":1,
  "method":"author_insertKey",
  "params":[
    "babe",
    "own word vocal dog decline set bitter example forget excite gesture water//1//babe",
    "0x48640c12bc1b351cf4b051ac1cf7b5740765d02e34989d0a9dd935ce054ebb21"
  ]
}
```

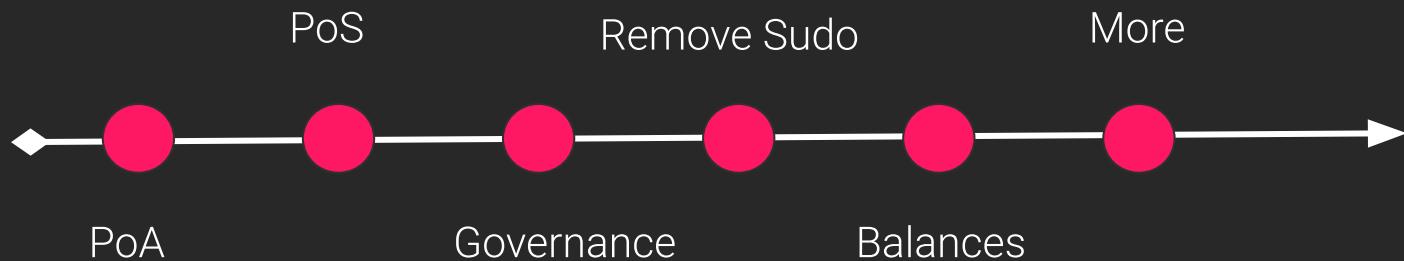
# Chain Specification

---

## 参考文档

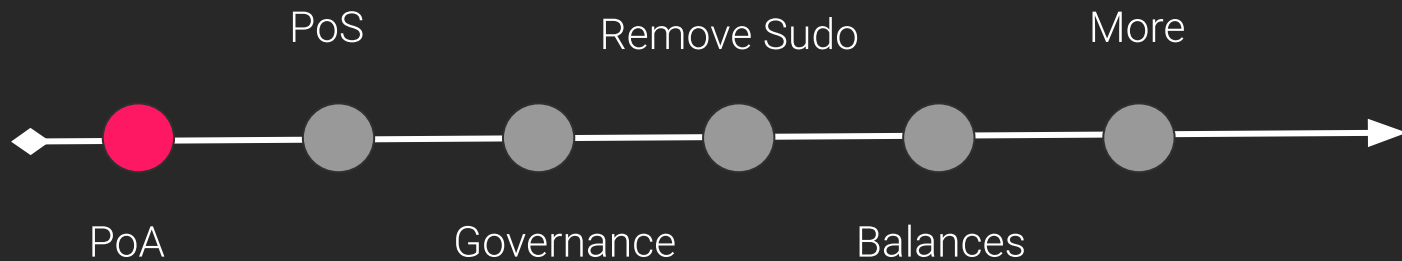
: <https://substrate.dev/docs/en/tutorials/start-a-private-network/customchain#option-2-use-curl>

# 部署公开测试网络



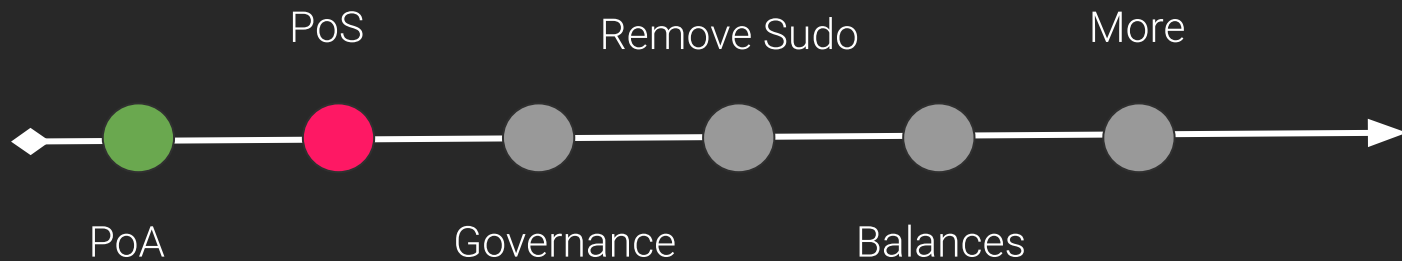
- 监控: <https://substrate.dev/docs/en/tutorials/visualize-node-metrics/>
- 编译WASM, srtool: <https://gitlab.com/chevdor/srtool>

# 部署公开测试网络



- StakingConfig force\_era 设置为 **ForceNone**
- Runtime system trait **BaseCallFilter** 过滤掉非必须的功能模块
- 允许提名和验证意向, 等待充足的提名和验证人
- 网络稳定后, 使用 sudo 调用 staking force\_new\_era 开启验证人选举

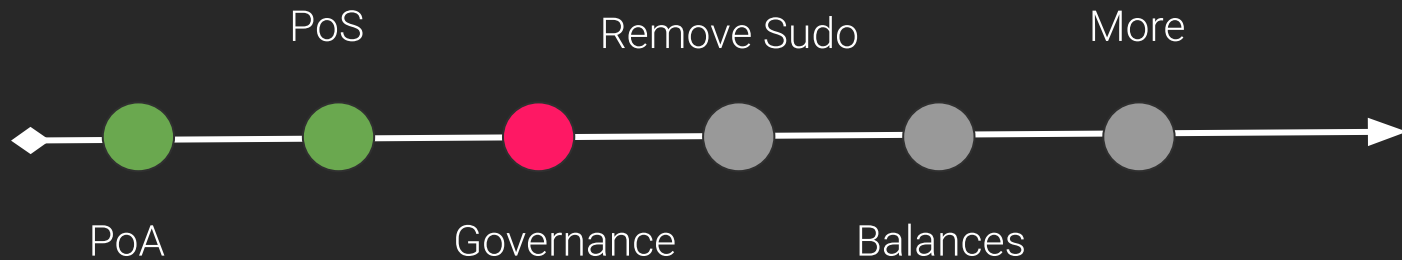
# 部署公开测试网络



- 根据网络情况, 调节验证人数量
- 使用 sudo 取消不必要的惩罚

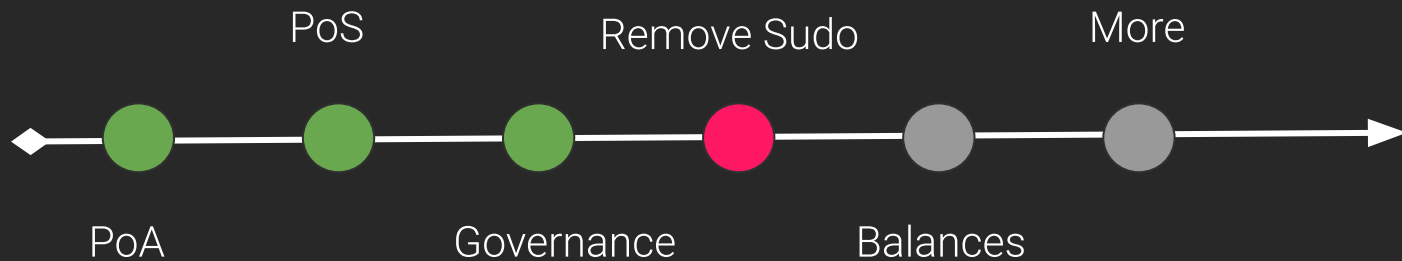


# 部署公开测试网络



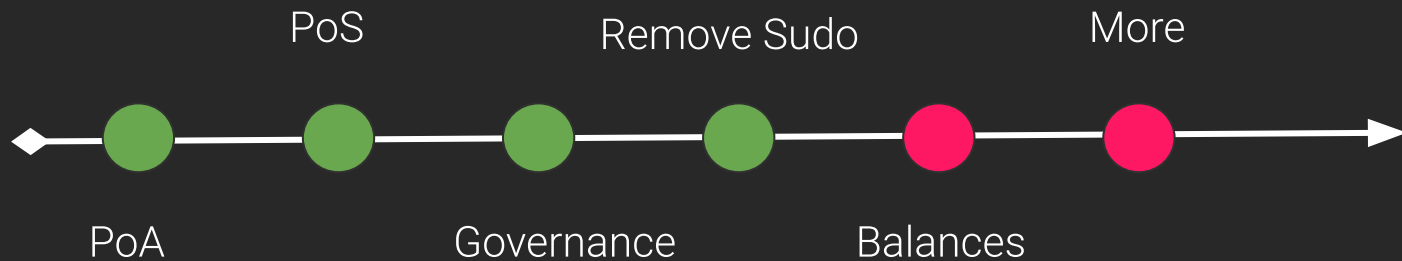
- 投票和选举议会成员
- 议会成员通过提案维护网络
- 通过提案使用国库资金
- 通过公投提案升级网络

# 部署公开测试网络



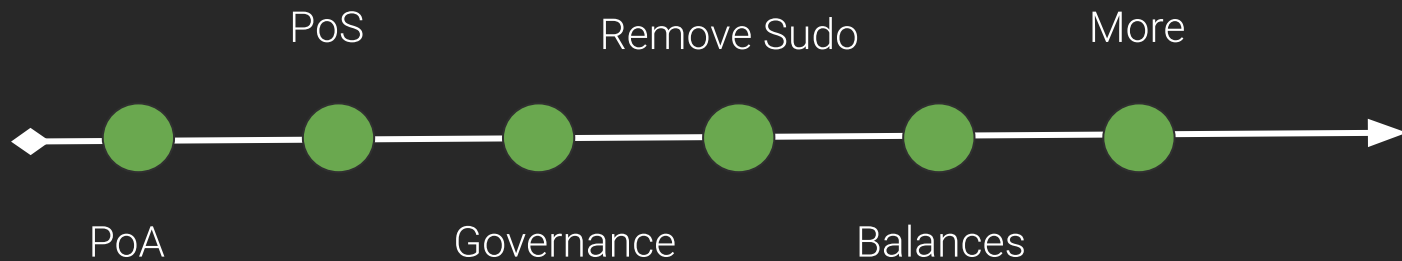
- 通过公投移除 sudo 权限

# 部署公开测试网络



- 打开转账功能, 及其它核心功能

# 部署公开测试网络



- runtime 升级数据迁移

# 安全审计

---

为什么要进行安全审计：

- 设计缺陷
- 代码 bug
- 错误的配置
- 出错后，难以修复且代价巨大
- 代码开源，供全世界范围审查

# 安全审计

---

## 原则：

- 多方、独立审计
- 上线前进行，保留充分的审计和修复时间
- 定期对新修改进行审计

## 方式：

- 静态检查
- 单元测试
- 手动 review
- 形式化验证

# 安全审计

---

Substrate 应用链的安全审计需要做哪些：

- 数值操作，如上溢、下溢
- 潜在的拒绝服务攻击
- 权重和交易费用的设置
- 随机数设计
- 异常处理
- 资源消耗
- .....

# 通证经济模型

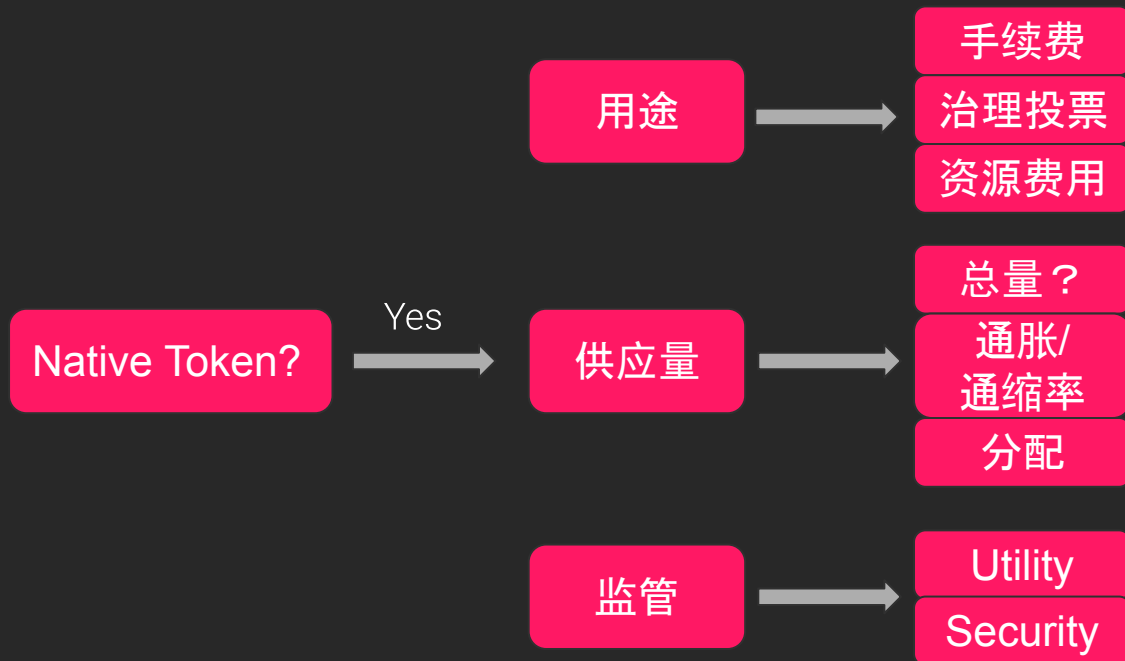
---

合理的通证设计,

- 作为网络中资源的 **所有权、使用权** 凭证
- 激励 **开源软件 / 开放网络** 的开发者、维护者
- **激励用户、参与治理**, 推动网络的使用和成长



# 通证经济模型



# 作业

---

- 为 template 模块的 do\_something 添加 benchmark 用例(也可以是其它自选模块的可调用函数), 并且将 benchmark 运行的结果转换为对应的权重定义;
- 选择 node-template 或者其它节点程序, 生成 Chain Spec 文件(两种格式都需要);
- (附加题)根据 Chain Spec, 部署公开测试网络

# 总结和展望

---

## 知识点：

- 区块链基本概念
- Substrate 开发知识和部署流程
- 智能合约和 Offchain worker
- 实战经验

# 总结和展望

---

接下来呢？

- Substrate 2.0 趋于稳定
- 应用链的正式上线和大范围应用
- 企业级联盟链的特性和用户案例
- 生态工具的完善
- 跨链消息传递的实现等

# Questions?

---

官网文档: [substrate.io](https://substrate.io)

知乎专栏: [parity.link/zhihu](https://parity.link/zhihu)

[kaichao@parity.io](mailto:kaichao@parity.io)

kaichaosun