经济系统--gas动态抵押算法

1.经济系统gas

目前经济系统相关的gas模型，是抵押COCOS获取gas的模型，没有衡量CPU、网络、存储硬盘等资源。

1.1交易费用

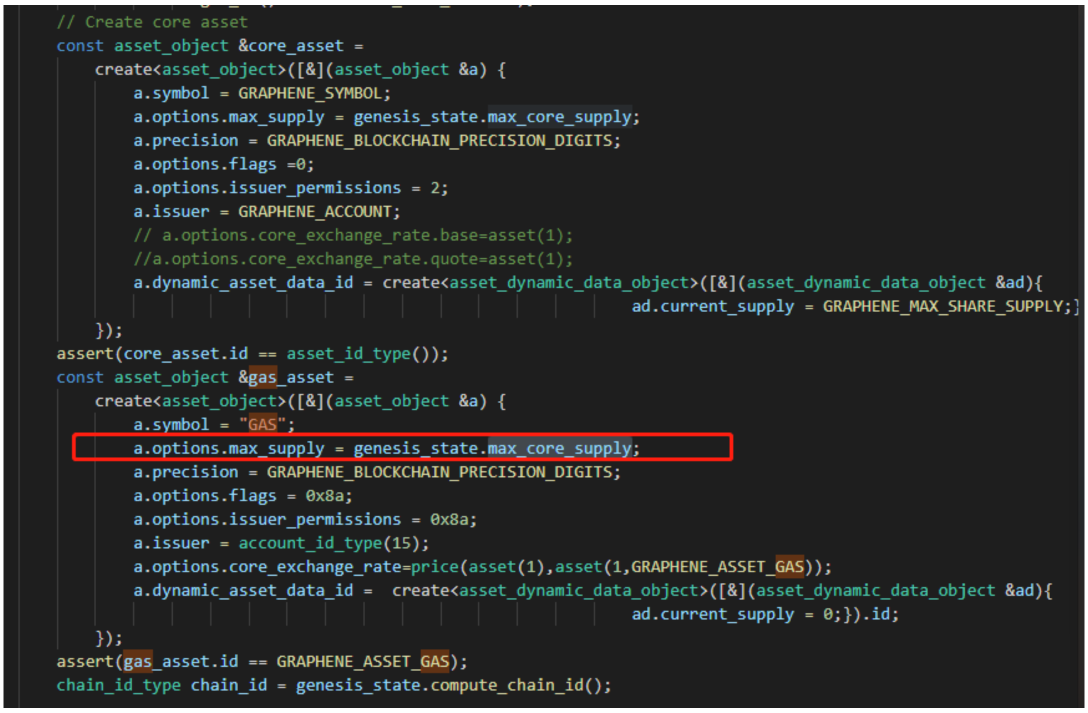
目前交易费用有两种方式：

- 抵押COCOS，获取gas，消耗gas；消耗的gas会线性回流
- gas不足，直接燃烧COCOS

两种方式，第二种方式直接按1:1比例消耗COCOS，简单明了；第一种方式需要测试cocos抵押gas的动态抵押算法。

1.2gas最大供应量配置

gas最大供应量默认与COCOS发行量一致



```
    // Create core asset
    const asset_object &core_asset =
        create<asset_object>([&](asset_object &a) {
            a.symbol = GRAPHENE_SYMBOL;
            a.options.max_supply = genesis_state.max_core_supply;
            a.precision = GRAPHENE_BLOCKCHAIN_PRECISION_DIGITS;
            a.options.flags =0;
            a.options.issuer_permissions = 2;
            a.issuer = GRAPHENE_ACCOUNT;
            // a.options.core_exchange_rate.base=asset(1);
            //a.options.core_exchange_rate.quote=asset(1);
            a.dynamic_asset_data_id = create<asset_dynamic_data_object>([&](asset_dynamic_data_object &ad){
                                                                ad.current_supply = GRAPHENE_MAX_SHARE_SUPPLY;}
        });
    assert(core_asset.id == asset_id_type());
    const asset_object &gas_asset =
        create<asset_object>([&](asset_object &a) {
            a.symbol = "GAS";
            a.options.max_supply = genesis_state.max_core_supply;
            a.precision = GRAPHENE_BLOCKCHAIN_PRECISION_DIGITS;
            a.options.flags = 0x8a;
            a.options.issuer_permissions = 0x8a;
            a.issuer = account_id_type(15);
            a.options.core_exchange_rate=price(asset(1),asset(1,GRAPHENE_ASSET_GAS));
            a.dynamic_asset_data_id =  create<asset_dynamic_data_object>([&](asset_dynamic_data_object &ad){
                                                                ad.current_supply = 0;}).id;
        });
    assert(gas_asset.id == GRAPHENE_ASSET_GAS);
    chain_id_type chain_id = genesis_state.compute_chain_id();
```

2.测试当前gas供应量

2.1当前gas供应量为0

```
unlocked >>> get_object 2.3.1
get_object 2.3.1
[{
    "id": "2.3.1",
    "current_supply": 0,
    "accumulated_fees": 0
```

```
 }
]
unlocked >>>
unlocked >>> list_account_balances 1.2.16
list_account_balances 1.2.16
4999999986164.97629 COCOS
```

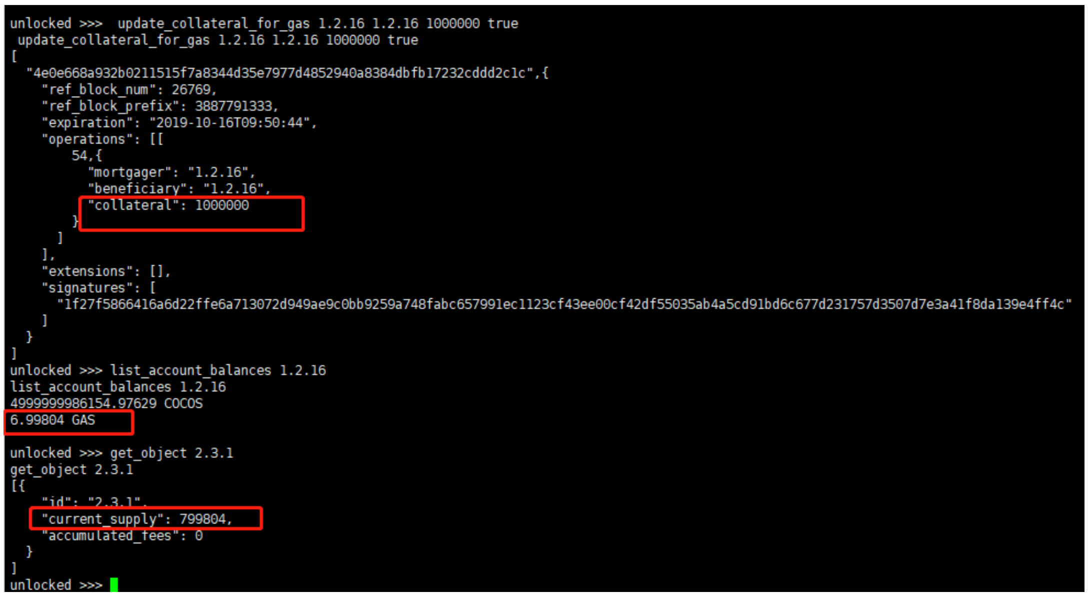
2.2抵押COCOS获取gas

```
unlocked >>>  update_collateral_for_gas 1.2.16 1.2.16 1000000 true
 update_collateral_for_gas 1.2.16 1.2.16 1000000 true
[
  "4e0e668a932b0211515f7a8344d35e7977d4852940a8384dbfb17232cddd2c1c",{
    "ref_block_num": 26769,
    "ref_block_prefix": 3887791333,
    "expiration": "2019-10-16T09:50:44",
    "operations": [[
        54,{
          "mortgager": "1.2.16",
          "beneficiary": "1.2.16",
          "collateral": 1000000
        }
      ]
    ],
    "extensions": [],
    "signatures": [
      
"1f27f5866416a6d22ffe6a713072d949ae9c0bb9259a748fabc657991ec1123cf43ee00cf42df55035ab4a5cd9
1bd6c677d231757d3507d7e3a41f8da139e4ff4c"
    ]
  }
]
```

2.3查看当前供应量

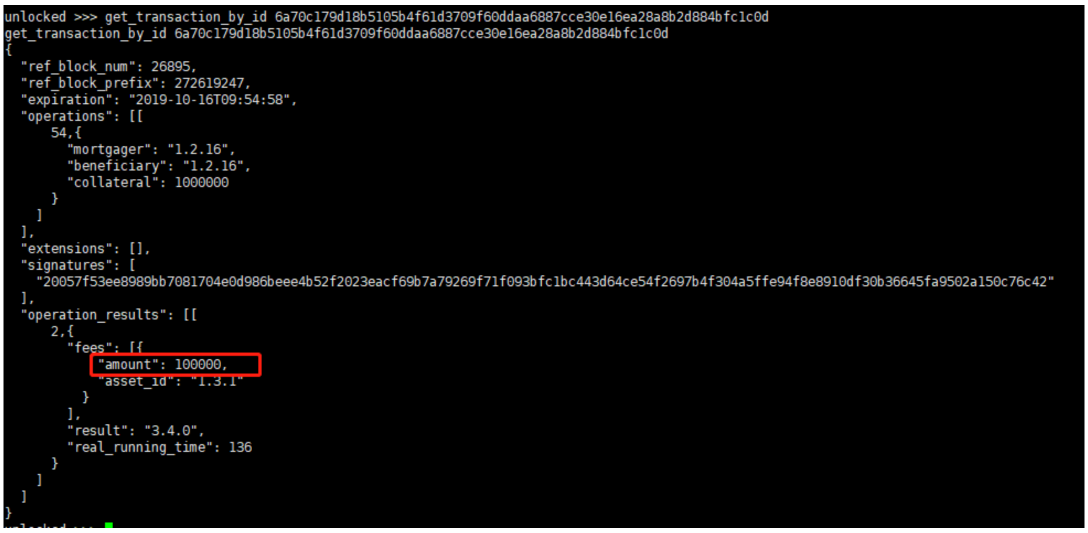
```
unlocked >>> list_account_balances 1.2.16
list_account_balances 1.2.16
4999999986154.97629 COCOS
6.99804 GAS
unlocked >>> get_object 2.3.1
get_object 2.3.1
[{
    "id": "2.3.1",
    "current_supply": 799804,
    "accumulated_fees": 0
  }
]
unlocked >>> get_transaction_by_id 
6a70c179d18b5105b4f61d3709f60ddaa6887cce30e16ea28a8b2d884bfc1c0d
get_transaction_by_id 6a70c179d18b5105b4f61d3709f60ddaa6887cce30e16ea28a8b2d884bfc1c0d
{
  ”ref_block_num”: 26895,
  ”ref_block_prefix”: 272619247,
  ”expiration”: ”2019–10–16T09:54:58”,
  ”operations”: [[
      54,{
          ”mortgager”: ”1.2.16”,
          ”beneficiary”: ”1.2.16”,
          ”collateral”: 1000000
      }
    ]
  ],
  ”extensions”: [],
  ”signatures”: [

”20057f53ee8989bb7081704e0d986beee4b52f2023eacf69b7a79269f71f093bfc1bc443d64ce54f2697b4f304
a5ffe94f8e8910df30b36645fa9502a150c76c42”
  ],
  ”operation_results”: [[
      2,{
          ”fees”: [{
              ”amount”: 100000,
              ”asset_id”: ”1.3.1”
            }
          ],
          ”result”: ”3.4.0”,
          ”real_running_time”: 136
      }
    ]
  ]
}
```



```
unlocked >>>  update_collateral_for_gas 1.2.16 1.2.16 1000000 true
 update_collateral_for_gas 1.2.16 1.2.16 1000000 true
[
  "4e0e668a932b0211515f7a8344d35e7977d4852940a8384dbfb17232cddd2c1c",{
    "ref_block_num": 26769,
    "ref_block_prefix": 3887791333,
    "expiration": "2019-10-16T09:50:44",
    "operations": [[
        54,{
          "mortgager": "1.2.16",
          "beneficiary": "1.2.16",
          "collateral": 1000000
        }
      ]
    ],
    "extensions": [],
    "signatures": [
      "1f27f5866416a6d22ffe6a713072d949ae9c0bb9259a748fabc657991ec1123cf43ee00cf42df55035ab4a5cd91bd6c677d231757d3507d7e3a41f8da139e4ff4c"
    ]
  }
]
unlocked >>> list_account_balances 1.2.16
list_account_balances 1.2.16
4999999986154.97629 COCOS
6.99804 GAS

unlocked >>> get_object 2.3.1
get_object 2.3.1
[{
    "id": "2.3.1",
    "current_supply": 799804,
    "accumulated_fees": 0
  }
]
unlocked >>>
```



```
unlocked >>> get_transaction_by_id 6a70c179d18b5105b4f61d3709f60ddaa6887cce30e16ea28a8b2d884bfc1c0d
get_transaction_by_id 6a70c179d18b5105b4f61d3709f60ddaa6887cce30e16ea28a8b2d884bfc1c0d
{
  "ref_block_num": 26895,
  "ref_block_prefix": 272619247,
  "expiration": "2019-10-16T09:54:58",
  "operations": [[
      54,{
        "mortgager": "1.2.16",
        "beneficiary": "1.2.16",
        "collateral": 1000000
      }
    ]
  ],
  "extensions": [],
  "signatures": [
    "20057f53ee8989bb7081704e0d986beee4b52f2023eacf69b7a79269f71f093bfc1bc443d64ce54f2697b4f304a5ffe94f8e8910df30b36645fa9502a150c76c42"
  ],
  "operation_results": [[
      2,{
        "fees": [{
            "amount": 100000,
            "asset_id": "1.3.1"
          }
        ],
        "result": "3.4.0",
        "real_running_time": 136
      }
    ]
  ]
}
```

可以看到当前1.2.16gas = current_supply – fee

3.动态抵押算法模型

3.1算法抽象

x= max supply gas

y= current supply gas

z= 抵押cocos

v= COCOS current supply

scale = z/v

gas_resrved = x – y

scale0 = (1+ scale)^0.4–1

amonut = gas_resrved * scale0

3.2算法精简

将amonut展开，去除中间层；并令w=amonut

x= max supply gas	定值，默认为100亿
y= current supply gas	一次操作抵押COCOS获取gas，current supply gas一致；多次操作y增加；即多次操作抵押，每次和前一次比较，current supply gas增加
z= 抵押的COCOS	
v= COCOS current supply	缓慢增加的一个数值，在一个短期时间段内，可以看成定值

w = (x – y) * ((1+ z/v)^0.4–1)

4.动态抵押算法需要满足的条件

1.抵押的COCOS（即z）越多，获得的gass（即w）越多；针对的是一次操作

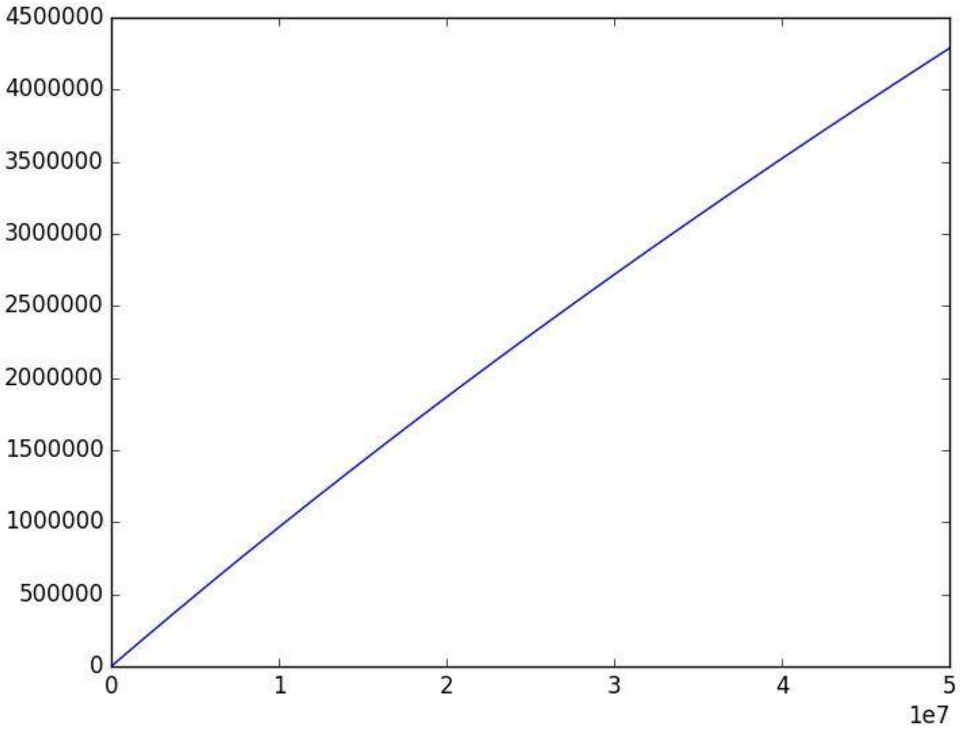
2.剩余gas越少，抵押COCOS获取gas越少；针对的是多次操作

5.测试

5.1测试条件1

一次操作，x、y、v固定；可以采用几个默认的固定值；w简化为:

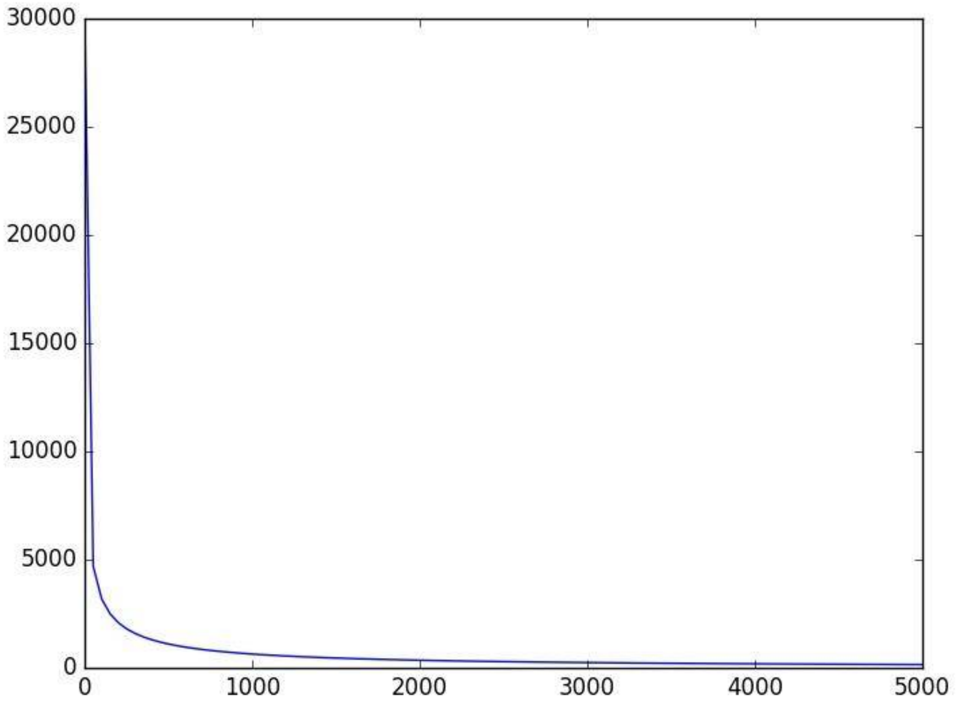
w = a * ((1+ z/b)^0.4–1)；a、b为常量；下图为相应的测试曲线：



可以看到这是一条递增的曲线，验证了条件1

5.2测试条件2

多次操作，此时有三个变量，为了将条件简化假设每次抵押的COCOS（即z）不变
x、y、z固定；可以采用几个默认的固定值；w简化为:
w = a * ((1+ b/v)^0.4−1)；a、b为常量；下图为相应的测试曲线：



可以看到这是一条递减的曲线，验证了条件2

6.附注

6.1测试代码1

```
import numpy as np
import matplotlib.pyplot as plt

x=np.linspace(1, 5000, 100)
y=2000 * ( (1+ x/8000)**0.4–1)
plt.figure()
plt.plot(x,y)
plt.savefig(”test4.jpg”)
```

6.2测试代码2

```
import numpy as np
import matplotlib.pyplot as plt

x=np.linspace(1, 5000, 100)
y=2000 * ( (1+ 1000/x)**0.4–1)
plt.figure()
plt.plot(x,y)
plt.savefig(”test5.jpg”)
```