1. 比较通用的要用到的函数：

core/db/DynamicPropertiesStore.java
存的是很多交易类型的交易费用，也就是下面的 fee= calcFee();
例如:AssetIssueActuator.java的调用链：

```java
public long getAssetIssueFee() {
return Optional.ofNullable(getUnchecked(ASSET_ISSUE_FEE))
.map(BytesCapsule::getData)
.map(ByteArray::toLong)
.orElseThrow(
() -> new IllegalArgumentException("not found ASSET_ISSUE_FEE"));
}
```

```java
try {
this.getAssetIssueFee();
} catch (IllegalArgumentException e) {
this.saveAssetIssueFee(1024000000L);
}
```

```java
public void setDynamicPropertiesStore(final DynamicPropertiesStore dynamicPropertiesStore) {
this.dynamicPropertiesStore = dynamicPropertiesStore;
}
```

```java
public DynamicPropertiesStore getDynamicPropertiesStore() {
return this.dynamicPropertiesStore;
}
```

这里的calcFee()不是一成变的，有些地方直接返回NULL，具体应该根据上下文代码判断

```java
public long calcFee() {
return dbManager.getDynamicPropertiesStore().getAssetIssueFee();
}
```
```java
long fee = calcFee();
```

总体来说貌似就是 blackhole 是___还没看懂

```java
/**
* Min TRX account.
*/
public AccountCapsule getBlackhole() {
return getUnchecked(assertsAddress.get("Blackhole"));
}
```

```java
byte[] getUnchecked(byte[] key);
```

这里的 getUnchecked 还有一种情况下 返回 NULL

```java
public T getUnchecked(byte[] key) {
byte[] value = revokingDB.getUnchecked(key);
```

```
try {
return of(value);
} catch (BadItemException e) {
return null;
}
}
```

2.各种能改变balance的类型:

(1)发布Token

java/org/tron/core/actuator/AssetIssueActuator.java

```
public long calcFee() {
return dbManager.getDynamicPropertiesStore().getAssetIssueFee();
}
```

```
dbManager.adjustBalance(ownerAddress, -fee);
//里面的逻辑就是直接运算
dbManager.adjustBalance(dbManager.getAccountStore().getBlackhole().getAddress().toByteArray(),
fee);//send to blackhole
```

(2)买储存?

java/org/tron/core/actuator/BuyStorageActuator.java

```
public long calcFee() {
return 0;
}
```

```
AccountCapsule accountCapsule = dbManager.getAccountStore()
.get(buyStorageContract.getOwnerAddress().toByteArray());
long quant = buyStorageContract.getQuant();

storageMarket.buyStorage(accountCapsule, quant);
long newBalance = accountCapsule.getBalance() - quant;
accountCapsule.setBalance(newBalance);
```

(3)

java/org/tron/core/actuator/BuyStorageBytesActuator.java
与BuyStorageActuator一样

(4)创建账户

java/org/tron/core/actuator/CreateAccountActuator.java

```
dbManager.adjustBalance(accountCreateContract.getOwnerAddress().toByteArray(), -fee);
// Add to blackhole address
dbManager.adjustBalance(dbManager.getAccountStore().getBlackhole().createDbKey(), fee);
```

(5)创建交易所

java/org/tron/core/actuator/ExchangeCreateActuator.java

```java
public long calcFee() {
return dbManager.getDynamicPropertiesStore().getExchangeCreateFee();
}
```

```java
long newBalance = accountCapsule.getBalance() - fee;
accountCapsule.setBalance(newBalance);
dbManager.getAccountStore().put(accountCapsule.createDbKey(), accountCapsule);
dbManager.getDynamicPropertiesStore().saveLatestExchangeNum(id);

dbManager.adjustBalance(dbManager.getAccountStore().getBlackhole().createDbKey(), fee);
```

(6)给交易所注资

java/org/tron/core/actuator/ExchangeInjectActuator.java

```java
public long calcFee() {
return 0;
}
```

```java
long newBalance = accountCapsule.getBalance() - calcFee();
accountCapsule.setBalance(newBalance);
```

(6.5)在交易所交易

java/org/tron/core/actuator/ExchangeTransactionActuator.java

```java
public long calcFee() {
return 0;
}
```

```java
long newBalance = accountCapsule.getBalance() - calcFee();
accountCapsule.setBalance(newBalance);
```

(7)从交易所撤资

java/org/tron/core/actuator/ExchangeWithdrawActuator.java
这里token和balance混合运算 比较奇怪

```java
public long calcFee() {
return 0;
}
```

```java
long newBalance = accountCapsule.getBalance() - calcFee();

if (Arrays.equals(tokenID, "_".getBytes())) {
accountCapsule.setBalance(newBalance + tokenQuant);
} else {
accountCapsule.addAssetAmountV2(tokenID, tokenQuant, dbManager);
}

if (Arrays.equals(anotherTokenID, "_".getBytes())) {
accountCapsule.setBalance(newBalance + anotherTokenQuant);
} else {
```

```
accountCapsule.addAssetAmountV2(anotherTokenID, anotherTokenQuant, dbManager);
}
```

(8)冻结资产

java/org/tron/core/actuator/FreezeBalanceActuator.java

```
long newBalance = accountCapsule.getBalance() - freezeBalanceContract.getFrozenBalance();
accountCapsule.setBalance(newBalance);
```

(9)购买发行的Token

java/org/tron/core/actuator/ParticipateAssetIssueActuator.java

```
public long calcFee() {
return 0;
}
```

```
long balance = Math.subtractExact(ownerAccount.getBalance(), cost);
balance = Math.subtractExact(balance, fee);//安全的减法
ownerAccount.setBalance(balance);
```

```
cost = participateAssetIssueContract.getAmount();
byte[] toAddress = participateAssetIssueContract.getToAddress().toByteArray();
AccountCapsule toAccount = this.dbManager.getAccountStore().get(toAddress);
toAccount.setBalance(Math.addExact(toAccount.getBalance(), cost));
```

(10)转账

java/org/tron/core/actuator/TransferActuator.java

```
long amount = transferContract.getAmount();
byte[] toAddress = transferContract.getToAddress().toByteArray();
byte[] ownerAddress = transferContract.getOwnerAddress().toByteArray();
```

```
dbManager.adjustBalance(ownerAddress, -fee);
dbManager.adjustBalance(dbManager.getAccountStore().getBlackhole().createDbKey(), fee);
```

(11)转账发布的Token

java/org/tron/core/actuator/TransferAssetActuator.java

```
public long calcFee() {
return 0;
}
```

```
if (toAccountCapsule == null) {
toAccountCapsule = new AccountCapsule(ByteString.copyFrom(toAddress), AccountType.Normal,
dbManager.getHeadBlockTimeStamp());
dbManager.getAccountStore().put(toAddress, toAccountCapsule);

fee = fee + dbManager.getDynamicPropertiesStore().getCreateNewAccountFeeInSystemContract();
}
```

```
dbManager.adjustBalance(ownerAddress, -fee);
```

```
dbManager.adjustBalance(dbManager.getAccountStore().getBlackhole().createDbKey(), fee);
```

java/org/tron/core/actuator/UnfreezeAssetActuator.java
没找到

(12)解冻资产
java/org/tron/core/actuator/UnfreezeBalanceActuator.java
```
public long calcFee() {
return 0;
}
```

```
long oldBalance = accountCapsule.getBalance();

long unfreezeBalance = 0L;;

byte[] receiverAddress = unfreezeBalanceContract.getReceiverAddress().toByteArray();
//If the receiver is not included in the contract, unfreeze frozen balance for this account.
//otherwise,unfreeze delegated frozen balance provided this account.
if (!ArrayUtils.isEmpty(receiverAddress) && dbManager.getDynamicPropertiesStore().supportDR())
{
byte[] key = DelegatedResourceCapsule
.createDbKey(unfreezeBalanceContract.getOwnerAddress().toByteArray(),
unfreezeBalanceContract.getReceiverAddress().toByteArray());
DelegatedResourceCapsule delegatedResourceCapsule = dbManager.getDelegatedResourceStore()
.get(key);

AccountCapsule receiverCapsule = dbManager.getAccountStore().get(receiverAddress);

switch (unfreezeBalanceContract.getResource()) {
case BANDWIDTH:
unfreezeBalance = delegatedResourceCapsule.getFrozenBalanceForBandwidth();
delegatedResourceCapsule.setFrozenBalanceForBandwidth(0, 0);
receiverCapsule.addAcquiredDelegatedFrozenBalanceForBandwidth(-unfreezeBalance);
accountCapsule.addDelegatedFrozenBalanceForBandwidth(-unfreezeBalance);
break;
case ENERGY:
unfreezeBalance = delegatedResourceCapsule.getFrozenBalanceForEnergy();
delegatedResourceCapsule.setFrozenBalanceForEnergy(0, 0);
receiverCapsule.addAcquiredDelegatedFrozenBalanceForEnergy(-unfreezeBalance);
accountCapsule.addDelegatedFrozenBalanceForEnergy(-unfreezeBalance);
break;
default:
//this should never happen
break;
}
accountCapsule.setBalance(oldBalance + unfreezeBalance);
```

(13)提取奖励

java/org/tron/core/actuator/WithdrawBalanceActuator.java

```java
AccountCapsule accountCapsule = (Objects.isNull(getDeposit())) ? dbManager.getAccountStore().
get(withdrawBalanceContract.getOwnerAddress().toByteArray()) :
getDeposit().getAccount(withdrawBalanceContract.getOwnerAddress().toByteArray());
long oldBalance = accountCapsule.getBalance();
long allowance = accountCapsule.getAllowance();

long now = dbManager.getHeadBlockTimeStamp();
accountCapsule.setInstance(accountCapsule.getInstance().toBuilder()
.setBalance(oldBalance + allowance)
.setAllowance(0L)
.setLatestWithdrawTime(now)
.build());
```

(14)创建超级节点候选人

java/org/tron/core/actuator/WitnessCreateActuator.java

```java
public long calcFee() {
return dbManager.getDynamicPropertiesStore().getAccountUpgradeCost();
}

long cost = dbManager.getDynamicPropertiesStore().getAccountUpgradeCost();
dbManager.adjustBalance(witnessCreateContract.getOwnerAddress().toByteArray(), -cost);

dbManager.adjustBalance(this.dbManager.getAccountStore().getBlackhole().createDbKey(),
+cost);
```

core/witness/WitnessController.java
貌似还没有加入reaward代码？？

```java
// TODO add vote reward
// long reward = Math.round(sum.get() * this.manager.getDynamicPropertiesStore()
// .getVoteRewardRate());
//account.setBalance(account.getBalance() + reward);
//accountStore.put(account.createDbKey(), account);
```

```java
         account.createReadableString();

// TODO add vote reward
// long reward = Math.round(sum.get() * this.manager.getDynamicPropertiesStore()
//     .getVoteRewardRate());
//account.setBalance(account.getBalance() + reward);
//accountStore.put(account.createDbKey(), account);
```