门店地图看板——MobX性能问题

Created and last modified by 王文磊 on Feb 15, 2022

一、问题描述

模拟环境下的看板加载（初始化、搜索）非常缓慢（macbook pro A1502 intel-i5）

地图看板左侧门店表格绑定了由mobx维护了的一个树形结构数据，第一层城市（30+），第二层区县（200+），第三层门店（2500+），初步定位该数据依赖绑定存在性能问题

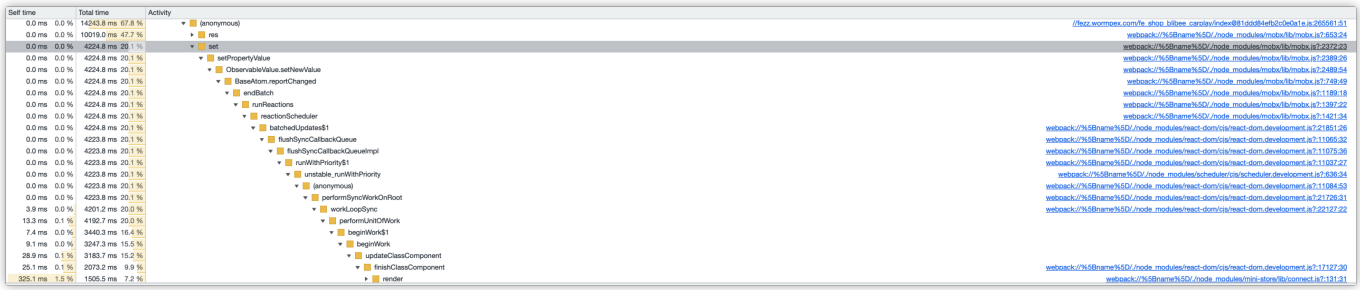
二、问题分析

工具：chrome performance

结论：通过performance下 call tree（根事件）分栏进行耗时分析，不难发现页面加载耗时最高的为微任务执行（即查询门店信息接口的 回调 / await 后的异步代码 ），查找调用链后发现时间主要花费在两处：

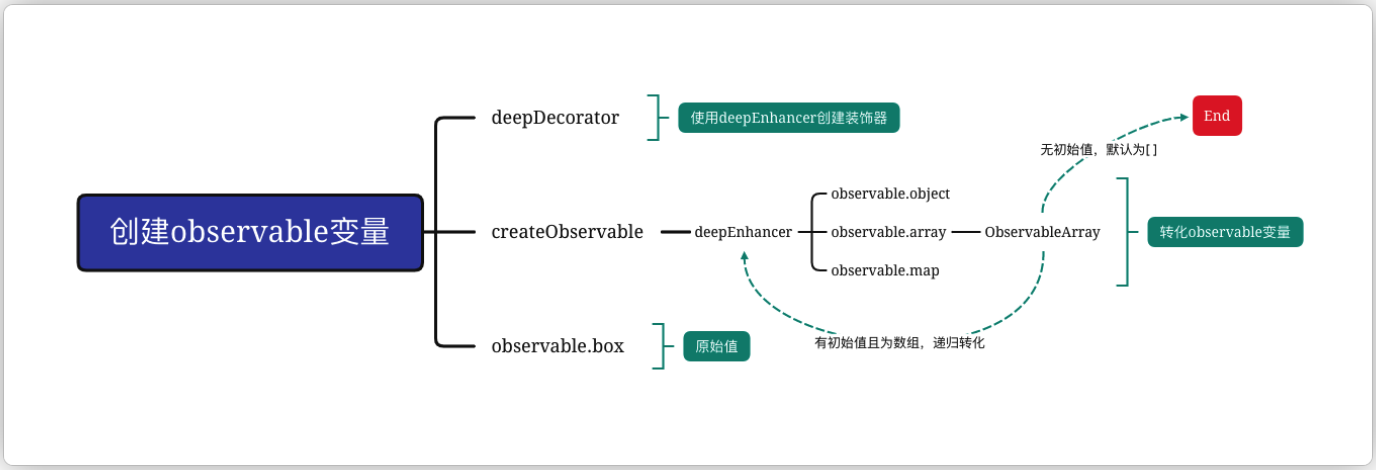
1. **Action中修改observable变量开启mobx批量事务处理（startBach），Action执行完成后关闭批量处理（endBatch），统一查找并执行Action中修改变量绑定的reaction（此处为 autorun 包裹的 render）后调用forceUpdate，触发 React updateClassComponent 组件更新机制**

2. **observable变量值变化触发被劫持的setter（mobx 3.x），将JS引用变量转化为observable引用变量的过程**



三、相关MobX源码 @3.1.9

Observable（以array为例）：

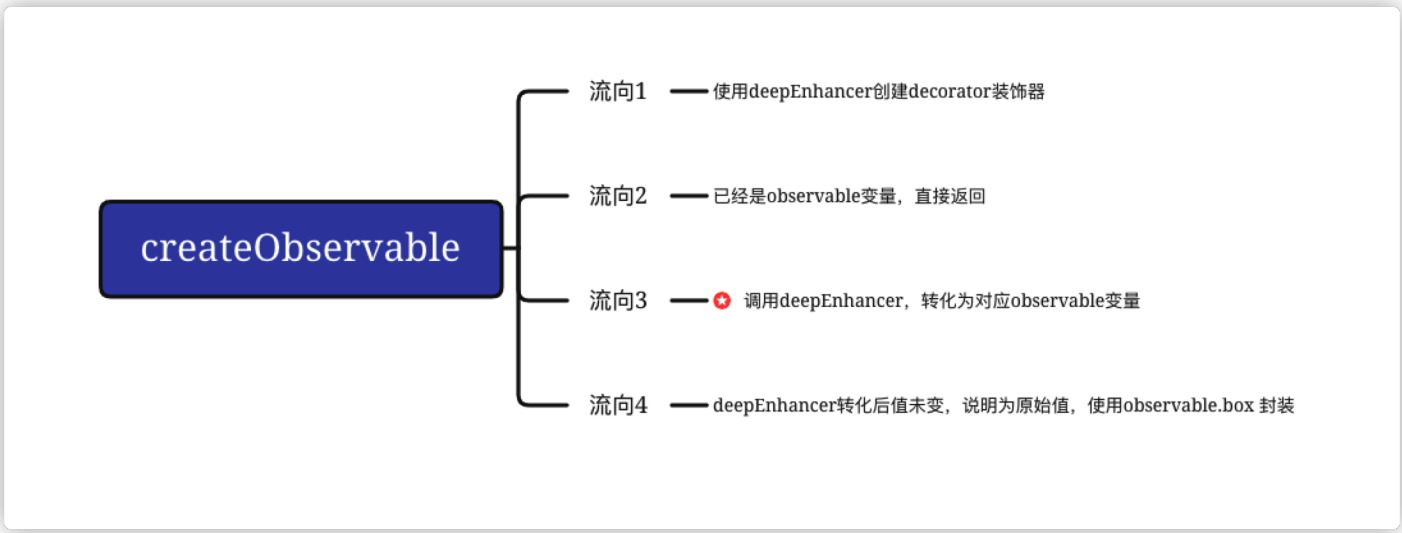


该函数本身不提供转换功能，只是起到 "转发" 作用，将传入的对象转发给对应具体的转换函数就行了；

createObservable

```
function createObservable(v) {
    if (v === void 0) { v = undefined; }
    if (typeof arguments[1] === "string")

        return deepDecorator.apply(null, arguments); // 流向1
    invariant(arguments.length <= 1, getMessage("m021"));
    invariant(!isModifierDescriptor(v), getMessage("m020"));
    if (isObservable(v))
        return v; // 流向2
    var res = deepEnhancer(v, undefined, undefined);
    if (res !== v)
        return res; // 流向3
    return observable.box(v); // 流向4
}
```



真正匹配类型，做相应转换

deepEnhancer

```
function deepEnhancer(v, _, name) {
    if (isModifierDescriptor(v))
        fail("You tried to assign a modifier wrapped value to a collection, please define mod
    if (isObservable(v))
        return v;
    if (Array.isArray(v))
        return observable.array(v, name);
    if (isPlainObject(v))
        return observable.object(v, name);
    if (isES6Map(v))
        return observable.map(v, name);
    return v;
}
```

observable上的这些方法是从 IObservableFactories 的原型对象上扩展来的：

```
exports.IObservableFactories = IObservableFactories;
var observable = createObservable;
exports.observable = observable;
Object.keys(IObservableFactories.prototype).forEach(function (key) { return observable[key] = IObservableFactories.prototype[key]; });
```

暴露给使用者的api工厂

IObservableFactories

```
var IObservableFactories = (function () {
    function IObservableFactories() {
    }
    IObservableFactories.prototype.box = function (value, name) {
        if (arguments.length > 2)
            incorrectlyUsedAsDecorator("box");
        return new ObservableValue(value, deepEnhancer, name);
    };
    IObservableFactories.prototype.shallowBox = function (value, name) {
        if (arguments.length > 2)
            incorrectlyUsedAsDecorator("shallowBox");
        return new ObservableValue(value, referenceEnhancer, name);
    };
    IObservableFactories.prototype.array = function (initialValues, name) {
        if (arguments.length > 2)
            incorrectlyUsedAsDecorator("array");
        return new ObservableArray(initialValues, deepEnhancer, name); // ---------------------
    };
    IObservableFactories.prototype.shallowArray = function (initialValues, name) {
        if (arguments.length > 2)
            incorrectlyUsedAsDecorator("shallowArray");
        return new ObservableArray(initialValues, referenceEnhancer, name);
    };
    IObservableFactories.prototype.map = function (initialValues, name) {
        if (arguments.length > 2)
            incorrectlyUsedAsDecorator("map");
        return new ObservableMap(initialValues, deepEnhancer, name);
    };
    IObservableFactories.prototype.shallowMap = function (initialValues, name) {
        if (arguments.length > 2)
            incorrectlyUsedAsDecorator("shallowMap");
        return new ObservableMap(initialValues, referenceEnhancer, name);
    };
    IObservableFactories.prototype.object = function (props, name) {
        if (arguments.length > 2)
            incorrectlyUsedAsDecorator("object");
        var res = {};
        asObservableObject(res, name);
        extendObservable(res, props);
        return res;
    };
    IObservableFactories.prototype.shallowObject = function (props, name) {
        if (arguments.length > 2)
            incorrectlyUsedAsDecorator("shallowObject");
        var res = {};
        asObservableObject(res, name);
        extendShallowObservable(res, props);
        return res;
    };
    IObservableFactories.prototype.ref = function () {
        if (arguments.length < 2) {
            return createModifierDescriptor(referenceEnhancer, arguments[0]);
        }
        else {
            return refDecorator.apply(null, arguments);
        }
```

```
    };
    IObservableFactories.prototype.shallow = function () {
        if (arguments.length < 2) {
            return createModifierDescriptor(shallowEnhancer, arguments[0]);
        }
        else {
            return shallowDecorator.apply(null, arguments);
        }
    };
    IObservableFactories.prototype.deep = function () {
        if (arguments.length < 2) {
            return createModifierDescriptor(deepEnhancer, arguments[0]);
        }
        else {
            return deepDecorator.apply(null, arguments);
        }
    };
    IObservableFactories.prototype.struct = function () {
        if (arguments.length < 2) {
            return createModifierDescriptor(deepStructEnhancer, arguments[0]);
        }
        else {
            return deepStructDecorator.apply(null, arguments);
        }
    };
    return IObservableFactories;
}());
```

通过 new ObservableArrayAdministration() 生成 observable array 实例，添加各种原型方法（以下省略），deep开启时（默认），使用deepEnhancer对数组每一项递归转化为observable属性，再将值赋予value

ObservableArray

```
function ObservableArray(initialValues, enhancer, name, owned) {
    if (name === void 0) { name = "ObservableArray@" + getNextId(); }
    if (owned === void 0) { owned = false; }
    var _this = _super.call(this) || this;
    var adm = new ObservableArrayAdministration(name, enhancer, _this, owned);
    addHiddenFinalProp(_this, "$mobx", adm);
    if (initialValues && initialValues.length) {
        adm.updateArrayLength(0, initialValues.length);
        adm.values = initialValues.map(function (v) { return enhancer(v, undefined, name + "[
        adm.notifyArraySplice(0, adm.values.slice(), EMPTY_ARRAY);
    }
    else {
        adm.values = [];
    }
    if (safariPrototypeSetterInheritanceBug) {
        Object.defineProperty(adm.array, "0", ENTRY_0);
    }
    return _this;
}
```

Like Be the first to like this