



提交阶段

- 1.布局计算 (Layout Calculation)
- 2.树提升, 从新树到下一棵树 (Tree Promotion, New Tree → Next Tree)
- 3.树对比 (Tree Diffing): 这个步骤会计算“先前渲染的树”(T)和“下一棵树”(T')的区别。计算的结果是原生视图的变更操作。(只在React 状态更新)

挂载阶段

- 1.树对比 (Tree Diffing) (只在初次渲染)
- 2.树提升
- 3.视图挂载 (View Mounting)

滑动或者操作和

观察各个阶段函数耗时发现:  
1.commit阶段最耗时在yoga布局 (LayoutTree函数中调用 YogaLayoutableShadowNode::YGNodeCalculateLayout)  
2.mount阶段最耗时在原生操作View (RCTPerformMountInstructions)

1.commit阶段耗时会影响渲染时间, 也有可能影响操作流畅度 (除了点击新增, 修改节点, 例如 QRN的slider 滑动, js拼接的 view。RN的slider滑动不影响 (原生组件UISlider直接滑动))  
2.mount阶段主线程调用函数, 会影响操作流畅度 (例如RN的 slider 滑动)

1.completeRoot调用多次?  
2.commit还由updateState调用