https://www.bilibili.com/video/BV1F4411D7rP

[Google Flutter 团队出品] Flutter 的性能测试和理论 (GDD China '18)

如何提高build效率

一、降低遍历出发点

更新的开始:

- 1.setState
- 2.InheritedWidget
- 3.热重载

调试工具

Performance Debugging Tools

- debugPrintBeginFrameBanner/debugPrintEndFrameBanner
 - 每帧开始/结束
- debugPrintScheduleBuildForStacks
 - 。 为什么被构建
- debugPrintRebuildDirtyWidgets
 - 。 什么组件被重新构建了
- debugProfileBuildsEnabled
 - 在观测台里显示构建树

调试工具

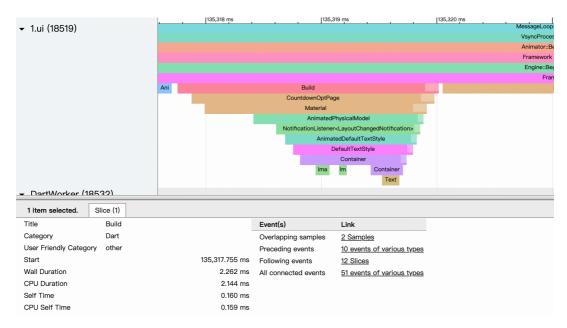
Performance Debugging Tools

- debugDumpLayerTree
 - 查看 layer 树
- debugPaintLayerBordersEnabled
 - 查看 layer 界限
- debugRepaintRainbowEnabled
 - o 被重新绘制的 RenderObject
- debugProfilePaintsEnabled
 - 。 在观测台里显示绘制树

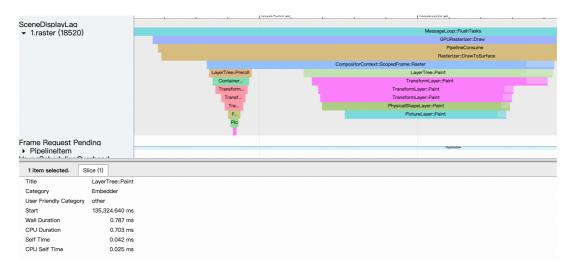
添加 debugProfileBuildsEnabled 可以在Observatory的Timeline看到Build过程 具体遍历了哪些节点

添加 debugProfilePaintsEnabled 可以在Observatory的Timeline看到paint过程 具体遍历了哪些节点

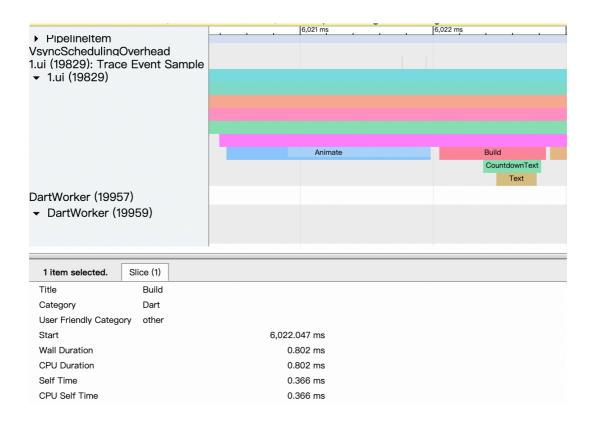
比如计时器,把Text拿出来,只更新Text文字,还要把Text用RepaintBoundary 隔离起来。如果只是把Text抽离成StatefulWidget,只能在build的时候减少了遍 历,但是在paint的时候还是全遍历

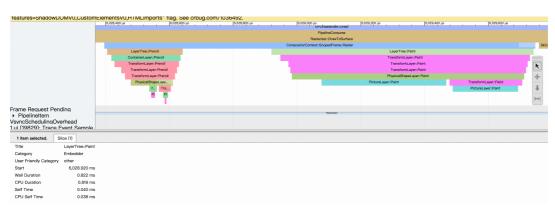


Build具体内容



Paint的具体内容







模拟器可以出来,真机出不来paint

二、停止树的遍历

```
AnimatedContainer(
curve: Curves.easeInOut,
duration: DurationConstants.ms200.
height: activeIndex == index ? side.px() : smallSide.px(),
width: activeIndex == index ? side.px() : smallSide.px(),
// color: activeIndex == index ? Colors.transparent : Colors.black12,
foregroundDecoration: activeIndex == index
? BoxDecoration(
color: Colors.transparent,
borderRadius: BorderRadius.circular(1000.px()),
: BoxDecoration(
color: ColorConstants.black16,
 borderRadius: BorderRadius.circular(1000.px()),
// margin: EdgeInsets.symmetric(vertical: 20.0, horizontal: 10.0),
child: ClipRRect(
borderRadius: BorderRadius.circular(1000.px()),
child: Stack(
alignment: Alignment.bottomCenter,
children: <Widget>[
ExtendedImage.network(
list[index].iconUrl,
width: side.px(),
height: side.px(),
fit: BoxFit.cover,
cache: true,
 gaplessPlayback: true,
loadStateChanged: (ExtendedImageState state) {
switch (state.extendedImageLoadState) {
case LoadState.loading:
```

return imageDefault();

