



Keishin Yokomaku / potatotips #41 @ Yahoo Japan, Inc.

AsyncLayoutInflater vs Litho

About Me

▶ Keishin Yokomaku

 Drivemode, Inc. / Principal Engineer

 Keith Yokoma: [GitHub](#) / [Twitter](#) / [Qiita](#) / [Tumblr](#) / [Stack Overflow](#)

▶ Books: [Android Academia](#) / [Grimoire of Android](#) / [Nanairo Android](#)

▶ Fun: Gymnastics / Cycling / Photography / Motorsport / Hiking

▶ Today's Quote: “富士山を自転車で登る爽快感”

AsyncLayoutInflater vs Litho

Overview of AsyncLayoutInflater

- ▶ Member of support v4 library.
- ▶ Inflate xml layout in a background thread.
 - ▶ To avoid blocking UI thread so that UI keeps being responsive.
 - ▶ Inflated layout is not added to parent.
- ▶ For more details: <https://goo.gl/prNKrl>.

Overview of Litho

- ▶ Declarative UI implementation framework.
- ▶ Supports asynchronous measure and layout.
 - ▶ Measure and layout operations run on a background thread.
 - ▶ Draw operation runs on UI thread.
- ▶ For more details: <https://goo.gl/Djkys0>.

How fast could they inflate the layout?

- ▶ Condition.
 - ▶ Put 10,000 TextViews in one ViewGroup and inflate it.
 - ▶ Tested with Nexus5X running Nougat 7.1.2.
- ▶ Code.
 - ▶ GitHub: <https://goo.gl/xGq9XT>.
 - ▶ Very rough implementation. 😊

Layout to inflate with AsyncLayoutInflater

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/scroll"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:layout_below="@+id/text"
9     tools:context="io.github.keithyokoma.asynclayout.MainActivity">
10     <LinearLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:orientation="vertical">
14         <TextView
15             android:layout_width="wrap_content"
16             android:layout_height="wrap_content"
17             android:text="Hello World!"/>
18         <TextView
19             android:layout_width="wrap_content"
20             android:layout_height="wrap_content"
21             android:text="Hello World!"/>
22         <TextView
23             android:layout_width="wrap_content"
24             android:layout_height="wrap_content"
25             android:text="Hello World!"/>
26         <TextView
27             android:layout_width="wrap_content"
28             android:layout_height="wrap_content"
29             android:text="Hello World!"/>
30         <TextView
31             android:layout_width="wrap_content"
32             android:layout_height="wrap_content"
33             android:text="Hello World!"/>
34         <TextView
35             android:layout_width="wrap_content"
36             android:layout_height="wrap_content"
37             android:text="Hello World!"/>
38         <TextView
39             android:layout_width="wrap_content"
40             android:layout_height="wrap_content"
41             android:text="Hello World!"/>
42         <TextView
43             android:layout_width="wrap_content"
44             android:layout_height="wrap_content"
45             android:text="Hello World!"/>
46         <TextView
47             android:layout_width="wrap_content"
48             android:layout_height="wrap_content"
49             android:text="Hello World!"/>
50         <TextView
51             android:layout_width="wrap_content"
52             android:layout_height="wrap_content"
53             android:text="Hello World!"/>
54         <TextView
55             android:layout_width="wrap_content"
56             android:layout_height="wrap_content"
57             android:text="Hello World!"/>
```


Layout to inflate with Litho

```
1 package io.github.keithyokoma.litho;
2
3 import com.facebook.litho.Column;
4 import com.facebook.litho.ComponentContext;
5 import com.facebook.litho.ComponentLayout;
6 import com.facebook.litho.annotations.LayoutSpec;
7 import com.facebook.litho.annotations.OnCreateLayout;
8 import com.facebook.litho.widget.Text;
9 import com.facebook.yoga.YogaDirection;
10 import com.facebook.yoga.YogaJustify;
11 import com.facebook.yoga.YogaWrap;
12
13 /**
14  * Created by KeishinYokomaku on 2017/06/05.
15  */
16 @LayoutSpec
17 /* package */ class HomeComponentSpec {
18     @OnCreateLayout
19     /* package */ static ComponentLayout onCreateLayout(ComponentContext c) {
20         ComponentLayout.ContainerBuilder builder = Column.create(c)
21             .justifyContent(YogaJustify.SPACE_AROUND)
22             .layoutDirection(YogaDirection.INHERIT)
23             .wrap(YogaWrap.NO_WRAP);
24         for (int i = 0; i < 10000; i++) {
25             builder = builder.child(Text.create(c)
26                 .text("Hello World!")
27                 .textSizeSp(14)
28                 .withLayout()
29                 .build());
30         }
31         return builder.build();
32     }
33 }
34
```


Result...

At first: No Contest

- ▶ No chance to win for Litho because it is too slow to compose a huge layout.
 - ▶ As [ProAndroidDev](#) mentioned, Litho is not a full replacement for XML.
- ▶ Problem in the implementation
 - ▶ No scrollable container for 10,000 texts so that Litho will squeeze them in bounds of screen size.

Litho and scrollable container

- ▶ RecyclerView
 - ▶ Of course it's scrollable.
- ▶ ScrollView
 - ▶ No equivalent component in Litho so far.
 - ▶ Parent view of LithoView can be ScrollView 😊

Again...

Result table

#	AsyncLayoutInflater	Litho
1	2,584,304,320 nano secs (2 secs)	19,826,206,923 nano secs (19 secs)
2	6,367,538,759 nano secs (6 secs)	21,579,648,348 nano secs (21 secs)
3	3,169,053,024 nano secs (3 secs)	22,534,724,276 nano secs (22 secs)
4	2,852,954,503 nano secs (2 secs)	22,701,877,992 nano secs (22 secs)
5	3,020,239,467 nano secs (3 secs)	21,354,948,117 nano secs (21 secs)
6	2,869,514,244 nano secs (2 secs)	21,115,173,145 nano secs (21 secs)
7	2,449,029,515 nano secs (2 secs)	21,619,548,768 nano secs (21 secs)
8	6,719,644,993 nano secs (6 secs)	19,353,791,876 nano secs (19 secs)
9	5,927,050,643 nano secs (5 secs)	20,020,680,380 nano secs (20 secs)
10	6,206,141,191 nano secs (6 secs)	20,739,572,795 nano secs (20 secs)
Ave	4216547065.9 nano secs (4 secs)	21084617262 nano secs (21 secs)

**AsyncLayoutInflater
is faster than Litho**



Keishin Yokomaku / potatotips #41 @ Yahoo Japan, Inc.

AsyncLayoutInflater vs Litho

We are hiring Android/iOS engineers!



We are hiring Android/iOS engineers!

