



Renderscript

Sylvain Galand

OPEN
WORLD
FORUM





Rendu ?

Scripts ?

Ni l'un, ni l'autre!



Embellir Android !

- Animations, fonds d'écran animés, animation de démarrage...

Un outil contre la "diversification"

- "Write once, run anywhere"
- Tous les GPUs (extensions OpenGL)
- Introduction en interne dans Android 2.x

Evolution vers les calculs

- Exécution optimale partout.
- Publique avec Android 3.x

Dépréciation de la partie rendu graphique

- Annoncée avec Jelly Bean (Android 4.1)

Une note dans la documentation

- Deprecation Notice: **Earlier versions of Renderscript** included an **experimental graphics engine** component. This component is now deprecated as of Android 4.1 [...] If you have apps that render graphics with Renderscript, we **highly** recommend you **convert your code to another Android graphics rendering option**.

Pourquoi ce changement ?

- Les développeurs préfèrent OpenGL
- OpenGL plus utilisé en interne aussi.

Un outil pour les calculs

- Ne remplace ni le SDK, ni le NDK.
- Répond à des problématiques spécifiques

Portabilité

- Android 3.0 +

Performance

- CPU (architecture, coeurs, instructions)
- GPU, bientôt... (ARM Mali-T604 GPU)

Simplicité d'utilisation

- Génération de code

Code: script.rs

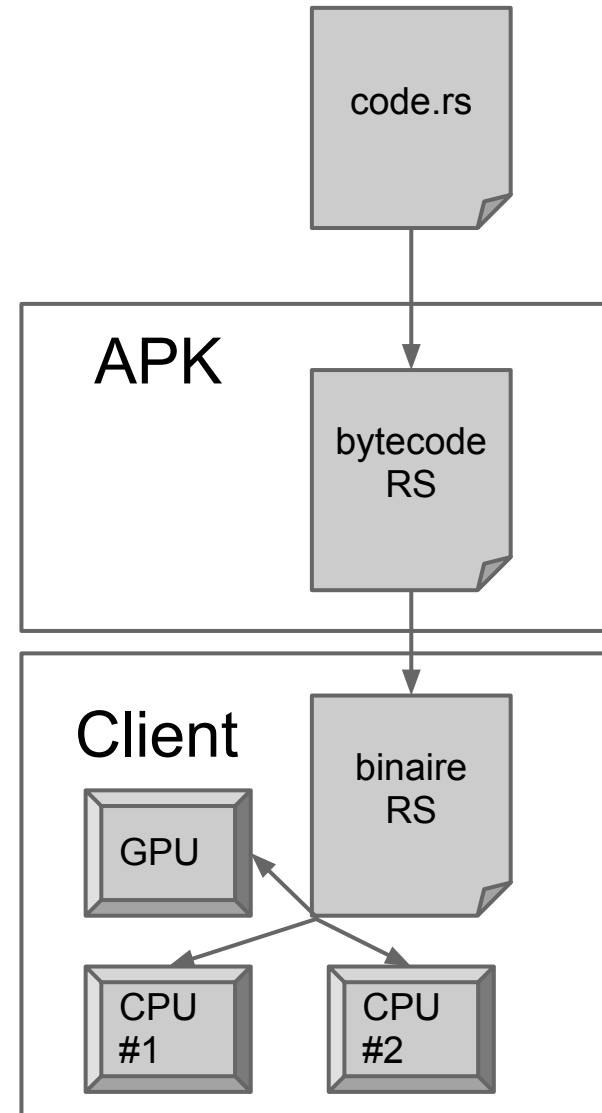
- Langage C99
- API pour le calculs

Compilations via LLVM

- Compilation en bytecode
- Compilation (JIT) sur le client

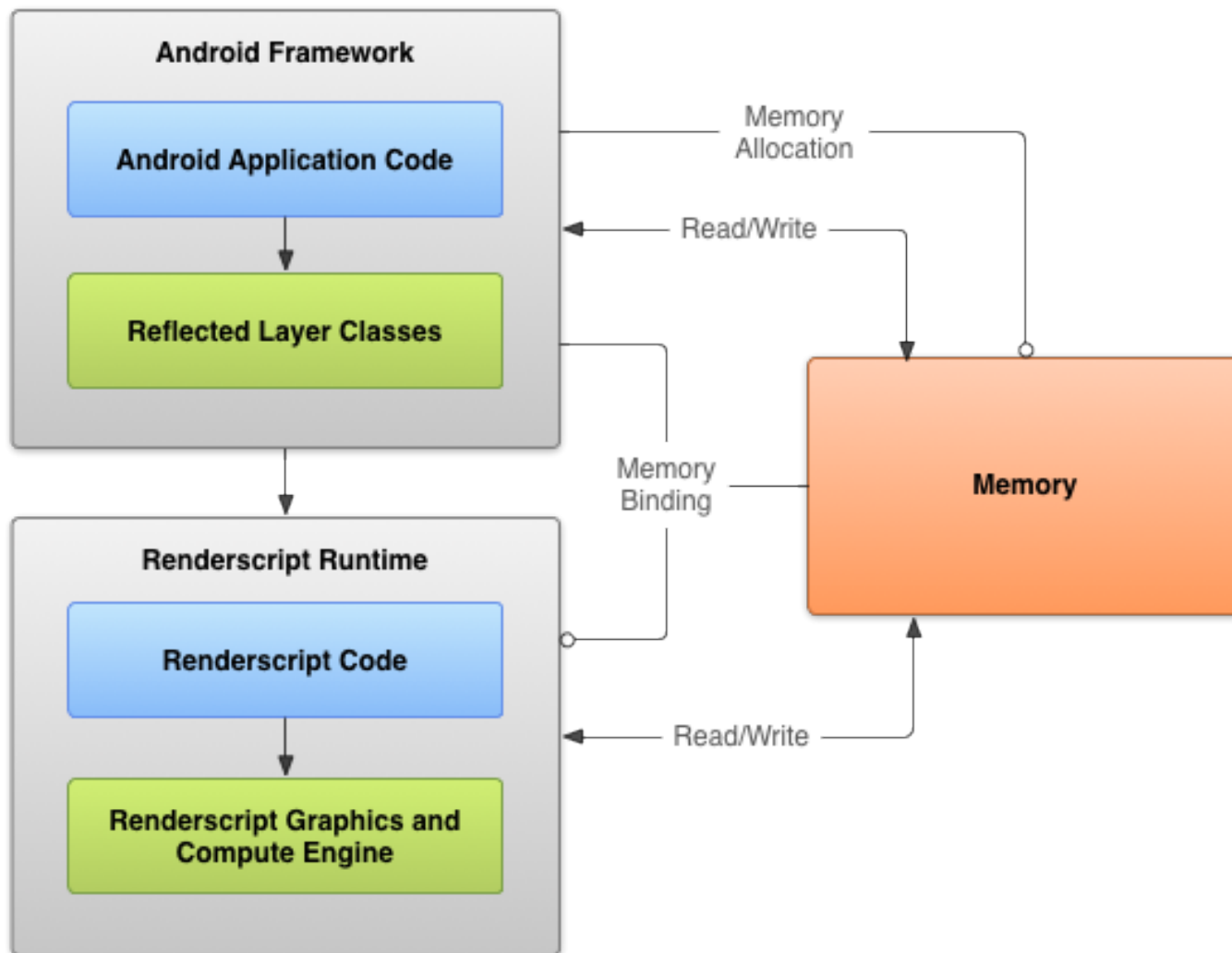
Execution partout

- Runtime Renderscript
- Coeurs CPU + GPU



Génération de "glue code"

Renderscript	Java
my_script.rs	ScriptC_my_script.java /res/raw/my_script.bc
typedef struct My_Struct	ScriptField_My_Struct.java
My_Struct_t* my_struct;	.bind_my_struct(...);
int entier;	.set_entier(int i) .get_entier();
void my_function();	.invoke_my_function();



Par exemple, un exemple

- Application de filtres sur des photos/images
- Traitement du signal, calcul matriciel
- (Rappel : Cette application vaut dans les 1 milliards)

Un peu de code

- Améliorons les temps de calcul grâce à Renderscript !

```
#pragma version(1)

#pragma rs java_package_name(com.genymobile.owf)

rs_matrix3x3 filter;

void init() { {
    rsMatrixLoadIdentity(&filter);
}

void root(const uchar4 *in,  uchar4 *out) {
    float3 pixel = convert_float4(in[0]).rgb;
    pixel = rsMatrixMultiply(&filter, pixel);
    pixel = clamp(pixel, 0.f, 255.f);
    out->a = in->a;
    out->rgb = convert_uchar3(pixel);
}
```

```
// Création du script
mRS = RenderScript.create(mContext);
mScript = new ScriptC_filter(mRS, mContext.getResources(), R.raw.filter);

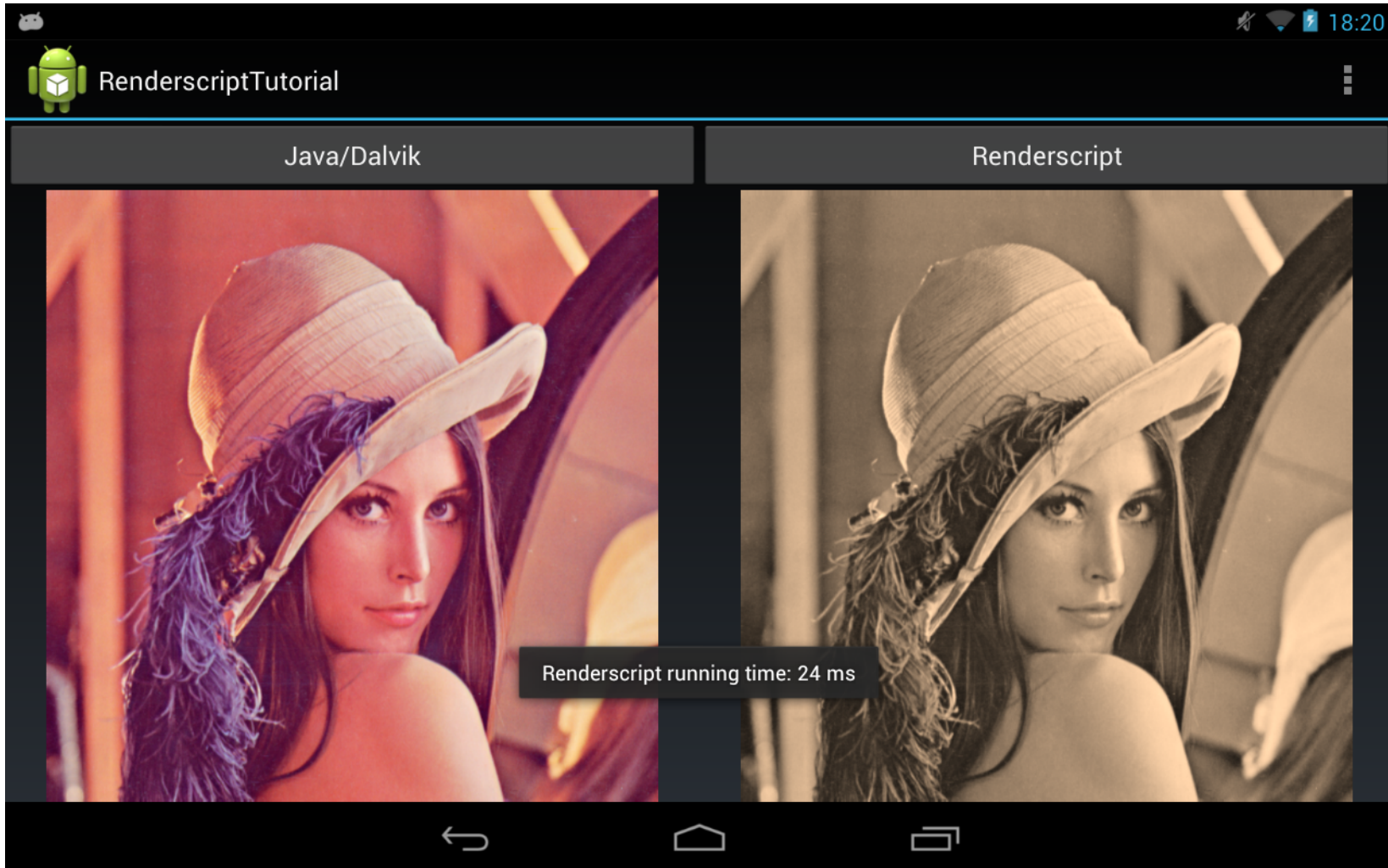
// Allocation de la mémoire contenant le bitmap
mInAllocation = Allocation.createFromBitmap( mRS,
    inputBitmap,
    Allocation.MipmapControl.MIPMAP_NONE,
    Allocation.USAGE_SCRIPT);

// Allocation de la mémoire pour récupérer l'image générée
mOutAllocation = Allocation.createTyped(mRS, mInAllocation.getType());

// Mise en place des paramètres
Matrix3f sepiaMatrix = new Matrix3f(mMatrix);
mScript.set_filter(sepiaMatrix);

// Appel du script (rsForEach)
mScript.forEach_root(mInAllocation, mOutAllocation);

// Copie du résultat dans le bitmap de sortie
mOutAllocation.copyTo(outputBitmap);
```



Appareil (matériel)	Java	RS	Gain
Nexus S (Exynos 1 GHz Cortex A8)	960 ms	280 ms	x 3.4
Galaxy Nexus (OMAP 4460 1,2 GHz dual-core)	360 ms	80 ms	x 4.5
Samsung Galaxy S2 (1.2 GHz dual-core ARM Cortex-A9)	340 ms	48 ms	x 7.0
Samsung Galaxy S3 (1.4 GHz quad-core ARM Cortex-A9)	325 ms	49 ms	x 6.6
Motorola Xoom (Tegra 2: 1GHz dual-core)	210 ms	26 ms	x 8.0
Nexus 7 (Tegra 3: 1.3 GHz quad-core)	180 ms	22 ms	x 8.2
Intel AZ210 - Intel Orange San Diego (Intel Atom 1.6 Ghz Z2460 with HT - x86)	293 ms	59 ms	x 4.9

seny mobile



Let IT be mobile

OPEN
WORLD
FORUM

Android 4.0.3+



En bref

- La solution pour les problèmes de performance sur les calculs
- Intégration simplifiée

Pour la suite

- Calculs dans le GPU ?
- Meilleurs outils ?
- Maintenance par Google ?

Merci !

- twitter : [@sylvaingaland](https://twitter.com/sylvaingaland)
- google plus : <http://slvn.fr/+>
- Code dispo sur : <http://github.com/sgaland>

