

# [SCA] DMS App 打包速度优化方案

- [Revision history](#)
  - [Ownership](#)
  - [Resources](#)
  - [Summary](#)
    - [Background](#)
    - [Goal](#)
  - [Detailed Design](#)
    - [Brand new](#)
      - 3.1、打包脚本支持并行打多个市场包
        - 3.1.1、多进程打多个市场包
        - 3.1.2、多进程对打包时间和系统负荷的影响
      - 3.2、DMS打包服务支持多机器并行打包
        - 3.2.1、打包脚本支持cid多选
        - 3.2.2、DMS打包服务支持多节点打包
      - 3.3、方案总结
      - 3.4、优化结果
- [Appendix](#)
- [MileStones](#)
- [Tickets](#)

## Revision history

Version	Revision date	Revisor	Revision content
v1	2022.05.11	jianwei	First draft

## Ownership

Native Dev	<a href="mailto:jianwei.luo@shopee.com">jianwei.luo@shopee.com</a>
Server Dev	<a href="mailto:xuanrui.zheng@shopee.com">xuanrui.zheng@shopee.com</a>
QA	TBD

## Resources

Git Repo(Optional)	<a href="https://git.garena.com/shopee/ssz-client/android/supplychain/corelib">https://git.garena.com/shopee/ssz-client/android/supplychain/corelib</a>
Project Schedule Page (Optional)	TBD

## Summary

### Background

Supply Chain移动端多个产品线App针对不同市场需要构建不同的包，随着新市场增加，SPX Driver App打全部市场的包需要构建9个，打包时间较长，会影响测试和发布效率。

另外目前Supply Chain移动端App打包全部在DMS平台，只有一台的8核16G内存的服务器在处理，因为服务器资源限制一次只能处理一个打包任务，如果多个打包任务需要排队，等待时间较长。

为了提高App打包速度，打包脚本需要对一次构建多个市场的App开启并行打包处理，提高单个任务的打包速度，并且DMS支持多个打包任务并行处理。

Goal

- 1. 打包脚本支持并行打多个市场包。
- 2. DMS支持多个打包任务并行处理。

Detailed Design

Brand new

3.1、打包脚本支持并行打多个市场包

Supply Chain 移动端的多市场包是通过Android 的 productFlavors 实现，使用 `./gradlew assembleRelease/Debug` 命令来构建全市场包、也可以使用 `./gradlew assembleIndonesiaRelease/Debug` 命令构建某个市场包。

```
android {
    productFlavors {
        indonesia {
            dimension 'default'
            buildConfigField('int', 'countryVersion', "${COUNTRY_VERSION_ID}")
            resConfigs 'in', 'ko'
        }
        malasia {
            dimension 'default'
            buildConfigField('int', 'countryVersion', "${COUNTRY_VERSION_MY}")
            resConfigs 'ms', 'ko'
        }
        ...
    }
}
```

市场差异代码深入到Java代码，无法通过 [Walle](#) 在Apk中的APK Signature Block区块添加自定义的市场信息来生成多市场包的方案进行提速。

还有个方案是开启多进程打包，每个进程打某个市场包，但能否开启9个进程并行打9个包呢？下面进行可行性分析

3.1.1、多进程打多个市场包

`./gradlew assembleRelease/Debug` 是使用串行方式来打多个市场包的，不支持并发打包，但是我们可以开启多进程打包，如果要开启3个进程，需要在3个代码工程分别构建3个包：

- 1. 进程1： `./gradlew assembleIndonesiaRelease assembleMalasiaRelease assembleThailandRelease --stacktrace`
- 2. 进程2： `./gradlew assemblePhilippinesRelease assembleVietnamRelease assembleSingaporeRelease --stacktrace`
- 3. 进程3： `./gradlew assembleTaiwanRelease assembleBrazilRelease assembleMexicoRelease --stacktrace`

那么理论上打包时间会是现在1/3，可以极大提升打包速度。实际可以开启多少个进程还需要根据系统资源来决定，所以需要对进程消耗资源做下分析。

3.1.2、多进程对打包时间和系统负荷的影响

并发多少进程需要根据服务器硬件资源来决定，因为并发进程太多，导致系统负载过高，造成服务器某些进程会被系统Kill，影响打包服务可靠性。下面我们分析下打包进程的资源消耗，为我们开启进程数量提供参考。

测试数据均在 DMS Live 以 SPX Driver APP v5.4.8版本进行测试，服务器配置：CPU：8核、内存：16G内存、磁盘类型：机械。

1、单进程打包

打包时间测试	进程1
测试1	6m 28s
测试2	6m 4s
测试3	5m 36s
平均	6m2s

开启一个进程情况下，CPU和内存还有余量，每个CPU核心平均负载在1以下，没有超负荷运行。

2、并发2个打包进程

打包时间测试	进程1	进程2
测试1	5m 1s	6m 15s
测试2	6m 3s	4m 57s
测试3	4m 41s	6m 17s
平均	5m 13s	5m 49s

```
jianwei.luo — root@sg2-shopee-cloud-devvm-test-10-143-253-200: ~ — ssh • ssh -i ~/.ssh/id_r1 10.143.253.200
...h • ssh -i ~/.ssh/id_r1 10.143.253.200 ...h -i ~/.ssh/id_r1 10.143.253.200 ...sh -i ~/.ssh/id_r1 10.143.253.200 +
top - 00:01:53 up 174 days, 7:36, 4 users, load average: 7.34, 6.80, 4.28
Tasks: 196 total, 1 running, 129 sleeping, 0 stopped, 0 zombie
%Cpu(s): 55.8 us, 2.5 sy, 0.5 ni, 41.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.1 st
KiB Mem : 15900372 total, 331048 free, 13612440 used, 1956884 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 1605308 avail Mem
```

开启两个打包进程的情况，打包速度和单进程差不多，没有变慢。但是1分钟内系统的平均负荷达到7.34，平均每个核心负荷： $7.34/8=0.9175$ ，接近满载了，并且内存可用内存仅剩300多MB。

3、并发3个进程

打包过程中，系统会随机Kill一个打包进程，导致测试中断了，gradle抛出异常信息：

```
FAILURE: Build failed with an exception.

* What went wrong:
Gradle build daemon disappeared unexpectedly (it may have been killed or may have crashed)

* Try:
Run with --info or --debug option to get more log output. Run with --scan to get full insights.

* Exception is:
org.gradle.launcher.daemon.client.DaemonDisappearedException: Gradle build daemon disappeared unexpectedly (it may have been killed or may have crashed)
    at org.gradle.launcher.daemon.client.DaemonClient.handleDaemonDisappearance(DaemonClient.java:250)
    at org.gradle.launcher.daemon.client.DaemonClient.monitorBuild(DaemonClient.java:223)
    at org.gradle.launcher.daemon.client.DaemonClient.executeBuild(DaemonClient.java:185)
    at org.gradle.launcher.daemon.client.DaemonClient.execute(DaemonClient.java:148)
    at org.gradle.launcher.daemon.client.DaemonClient.execute(DaemonClient.java:99)
    at org.gradle.launcher.cli.RunBuildAction.run(RunBuildAction.java:56)
```

```
jianwei.luo — root@sg2-shopee-cloud-devvm-test-10-143-253-200: ~ — ssh • ssh -i ~/.ssh/id_r1 10.143.253.200 — 110x33
...ssh/id_r1 10.143.253.200 ...ssh/id_r1 10.143.253.200 ...ssh/id_r1 10.143.253.200 .../ssh/id_r1 10.143.253.200 .../ssh/id_r1 10.143.253.200 +
top - 13:32:49 up 174 days, 21:06, 6 users, load average: 15.15, 10.25, 8.28
Tasks: 203 total, 1 running, 138 sleeping, 0 stopped, 0 zombie
%Cpu(s): 93.1 us, 6.2 sy, 0.7 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 15900372 total, 4876248 free, 9215068 used, 1809056 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 6006276 avail Mem
```

发现此时CPU全部跑满，1分钟内系统的平均负荷达到15，平均每个核心负荷： $15/8=1.875$ ，超过1了属于超负载运行了。

4、结论

综上测试数据，并不能根据打多少个市场包去并开启多少个打包进程，能开启多少进程受系统资源限制。

根据测试结果推断一个打包进程相对合理配置：4核CPU+8G内存，同时加上系统和DMS打包服务对资源的消耗情况（4核CPU+4G内存）

- 1. 开启1个进程需要 8核CPU+12G内存；
- 2. 开启3个进程需要 16核CPU+32G内存。

目前DMS Live 服务器配置仅有8核CPU+16G内存，不满足开启多进程。如果需要在单个虚拟机支持多进程，需要进行扩容。

### 3.2、DMS打包服务支持多机器并行打包

把多个CID的包均分到已有机器上面进行，例如，把all拆分到两个节点中运行：id,my,th,ph,vn和sg,tw,br,mx。需要对打包脚本和DMS打包服务进行改造：

- 1. 打包脚本支持cid多选；
- 2. DMS打包服务支持多节点打包。

#### 3.2.1、打包脚本支持cid多选

Gradle 命令支持多个命令进行串行执行，如果要支持多CID打包，可以使用下面命令：./gradlew assembleVietnamRelease assembleThailandRelease assembleBrazilRelease assembleIndonesiaRelease

其中第二个及之后的包构建会基于第一个包构建缓存进行增量编译，速度是第一个包的40%左右。

#### 3.2.2、DMS打包服务支持多节点打包

DMS打包服务支持多节点，由xuanrui.zheng负责出方案：[打包服务容器化技术方案](#)

### 3.3、方案总结

目前SRE已不再审批大容量虚拟机，选择多机器并行打包方案，也方便后续机器扩展。

### 3.4、优化结果

DMS Live 以 SPX Driver APP，CID选ALL，打release包测试：

打包时间测试	优化前	优化后（2个节点）
测试1	30m 20s	13m 59s
测试2	29m 53s	15m 9s
测试3	30m 6s	14m 43s
平均	30m 6s	14m 37s

使用2个节点，选ALL打9个包速度提升 51.4%。

## Appendix

- 1. [\[SCA\] Docker镜像Android打包脚本接口协议](#)

## MileStones

[\[SCA\] DMS App打包速度优化 Roadmap](#)

## Tickets

Task content	PIC	JIRA Ticket
构建打包速度优化	<a href="#">jianwei.luo</a>	<a href="#">SPSCTP-4062</a> - 正在获取问题细节。。。 <span>状态</span>