

# 附录 A: “基于 KML\_VML 的 JAVA 语言适配”技术合作项目 (SOW)

## 目录

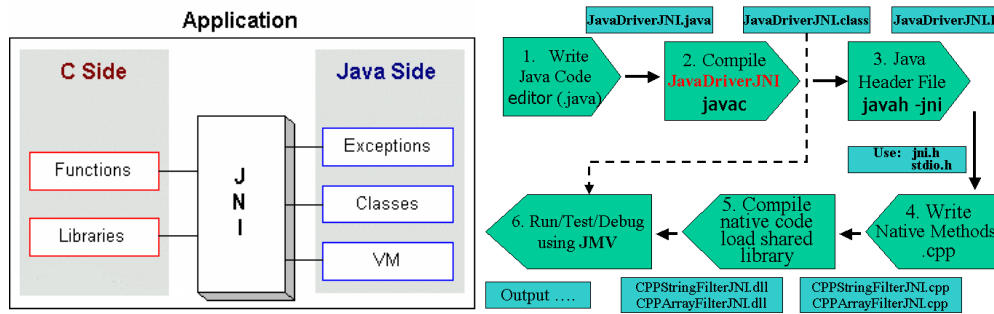
1.	合作需求/任务.....	2
1.1	研究领域.....	2
1.2	研究目标.....	2
1.4	其他要求 .....	4
2.	验收方法与指标 .....	4
2.1	验收要求: .....	4
3.	项目阶段.....	5
4.	项目监控和管理.....	5

# 1. 合作需求/任务

## 1.1 研究领域

基于鲲鹏处理器完成高性能向量运算函数库 Java 平台适配，本次合作包括基础运算、三角函数、指数对数等函数接口适配，总计约 32 个函数。

Java 语言通过 JNI 调用 C 接口开发高性能函数库基本原理如下：



## 1.2 研究目标

- (1) 基础运算函数，需求函数接口 8 个，根据数据类型和接口定义完成相关功能和优化。适配函数需求如下：

提供的 C 语言接口列表：

```
void vsadd(const int len, const float *src1, const float *src2, float *dst);
void vdadd(const int len, const double *src1, const double *src2, double *dst);
void vssub(const int len, const float *src1, const float *src2, float *dst);
void vdsb(const int len, const double *src1, const double *src2, double *dst);
void vsmul(const int len, const float *src1, const float *src2, float *dst);
void vdmul(const int len, const double *src1, const double *src2, double *dst);
void vsdiv(const int len, const float *src1, const float *src2, float *dst);
void vddiv(const int len, const double *src1, const double *src2, double *dst);
```

需实现的 Java 语言接口列表：

```
void vsAdd(int len, final float[] src1, final float[] src2, float dst);
void vdAdd(int len, final double[] src1, final double[] src2, float dst);
void vsSub(int len, final float[] src1, final float[] src2, float dst);
void vdSub(int len, final double[] src1, final double[] src2, float dst);
void vsMul(int len, final float[] src1, final float[] src2, float dst);
void vdMul(int len, final double[] src1, final double[] src2, float dst);
void vsDiv(int len, final float[] src1, final float[] src2, float dst);
void vdViv(int len, final double[] src1, final double[] src2, float dst);
```

- (2) 三角函数，需求函数接口 12 个，根据数据类型和接口定义完成相关功能和优化。适配函数需求如下：

提供的 C 语言接口列表：

```
void vsatan(const int len, const float *src, float *dst);
void vdatan(const int len, const double *src, double *dst);
void vsatan2(const int len, const float *src1, const float *src2, float *dst);
void vdatan2(const int len, const double *src1, const double *src2, double *dst);
void vscos(const int len, const float *src, float *dst);
void vdcos(const int len, const double *src, double *dst);
void vssin(const int len, const float *src, float *dst);
void vdsin(const int len, const double *src, double *dst);
void vssincos(const int len, const float *src, const float *sindst, float *cosdst);
```

```
void vdsincos(const int len, const double *src, const double *sindst, double *cosdst);
void vstan(const int len, const float *src, float *dst);
void vdtan(const int len, const double *src, double *dst);
```

需实现的 Java 语言接口列表:

```
void vsAtan(int len, final float[] src, float[] dst);
void vdAtan(int len, final double[] src, float[] dst);
void vsAtan2(int len, final float[] src1, final float[] src2, float[] dst);
void vdAtan2(int len, final double[] src1, final double[] src2, float[] dst);
void vsCos(int len, final float[] src, float[] dst);
void vdCos(int len, final double[] src, float[] dst);
void vsSin(int len, final float[] src, float[] dst);
void vdSin(int len, final double[] src, float[] dst);
void vsSinCos(int len, final float[] src, float[] sinDst, float[] cosDst);
void vdSinCos(int len, final double[] src, double[] sinDst, float[] cosDst);
void vsTan(int len, final float[] src, float[] dst);
void vdTan(int len, final double[] src, float[] dst);
```

- (3) 指数和对数函数, 需求函数接口 12 个; 根据数据类型和接口定义完成相关功能和优化。适配函数需求如下:

提供的 C 语言接口列表:

```
void vsexp(const int len, const float *src, float *dst);
void vdexp(const int len, const double *src, double *dst);
void vspow(const int len, const float *src1, const float *src2, float *dst);
void vdpow(const int len, const double *src1, const double *src2, double *dst);
void vslog10(const int len, const float *src, float *dst);
void vdlog10(const int len, const double *src, double *dst);
void vsln(const int len, const float *src, float *dst);
void vdln(const int len, const double *src, double *dst);
void vssqrt(const int len, const float *src, const float *dst);
void vdsqrt(const int len, const double *src, const double *dst);
void vssqr(const int len, const float *src, float *dst);
void vdsqr(const int len, const double *src, double *dst);
```

需实现的 Java 语言接口列表:

```
void vsExp(int len, final float[] src, float[] dst);
void vdExp(int len, final double[] src, float[] dst);
void vsPow(int len, final float[] src1, final float[] src2, float[] dst);
void vdPow(int len, final double[] src1, final double[] src2, float[] dst);
void vsLog10(int len, final float[] src, float[] dst);
void vdLog10(int len, final double[] src, float[] dst);
void vsLn(int len, final float[] src, float[] dst);
void vdLn(int len, final double[] src, float[] dst);
void vsSqrt(int len, final float[] src, float[] dst);
void vdSqrt(int len, final double[] src, double[] dst);
void vsSqr(int len, final float[] src, float[] dst);
void vdSqr(int len, final double[] src, float[] dst);
```

- (4) Java 接口适配的相关要求:

- 1、函数所在的包名: com.huawei.kml.vml
- 2、使用 Maven 进行项目代码管理;
- 3、能够一键式完成编译打包。

### 1.3 交付成果

#### 1) 软件交付

分阶段交付目标函数 32 个，见 1.2 描述。

交付软件需符合性能指标，如 2.1 描述。

交付对应函数的测试用例、测试数据样本。

#### 2) 文档交付

序号	交付
1	kml-java 向量计算函数 Java 适配软件架构设计说明书
2	kml-java 向量计算函数 Java 函数接口说明书 (按华为模版撰写)
3	kml-java 向量计算函数 Java 软件包测试报告

### 1.4 其他要求

- 1) 项目开发人员需有软件项目开发工作经验，熟悉C语言、Java语言，JNI框架。
- 2) 输出代码符合代码规范，代码注释英文化，重要的接口、核心功能部分必须要有注释说明。
- 3) 输出代码不得引用 GPLv2/v3、AGPL 的库文件或代码。
- 4) 使用的第三方软件清单列表，引入和刷新第三方软件要及时知会华为方并经过华为方同意。
- 5) 项目结束后，开发者需要继续六个月的bug维护期，维护期间对bug应做到2天内响应给出方案，1周内解决。
- 6) 故意放置恶意、安全漏洞代码的，将保留追究责任的一切权利。
- 7) 源代码必须经过业界主流静态扫描工具扫描，并清零。

## 2. 验收方法与指标

### 2.1 验收要求：

- 1) 功能对比验收，测试平台：Kunpen。

对比平台配置：

类别	TaiShan 200
CPU	Kunpeng 920 5250
核数	48
频率	2.6GHz（锁频）
内存大小	8G*6
操作系统	CentOS 7.6
页大小	64KB

#### 2) 功能指标

- 1) 完成各模块定义功能，对比 C 语言接口实现。
- 2) 函数输出需与 C 语言接口输出完全一致。

#### 3) 性能指标

- a) 对比 C 语言接口在 Kunpeng 平台上的性能指标。
- b) 所有接口性能需达到 C 语言接口性能的 95%以上。
- c) 测试方案：同一样本、同一测试用例分别调用 Java 接口和 C 接口，计算运行时，根据运行时间折算性能参数：

性能对比：Java 对比 C 性能 = (Java 运行时间) / (C 运行时间)

- d) 如以上指标无法达成，需给出优化结果分析和无法达成原因分析，双方评审确认。

- 4) 兼容性需求:
  - a) 支持编译器 OpenJDK1.8.0+GCC7.3.0(兼容 GCC4.8.5)
  - b) 兼容性测试基于 Centos 7.6。
  
- 5) 代码质量要求:
  - a) 编码需要满足华为要求，参考华为提供的编码规范。
  - b) 满足华为方静态扫描要求，静态检查问题清零。
  - c) 满足华为方安全要求，安全扫描问题清零，项目涉及使用的加密方案、算法、模块满足华为方要求。

### 3. 项目阶段

阶段	起始时间	任务描述	任务目标	交付成果
1 阶段	2021/1/01~ 2020/2/30	kml-java 向量计算函数技术方案设计	kml-java 向量计算函数功能方案、性能测试方案、测试方案设计，通过华为专家评审。	《鲲鹏数学库-向量计算函数 Java 适配设计说明书
2 阶段	2020/2/1~ 2021/5/30	kml-java 向量计算函数功能函数开发、测试及验收	kml-java 向量计算函数功能函数开发、文档编写、性能优化、功能和性能测试，提交函数代码、文档和测试报告，与华为研发人员一起完成验收工作。	kml-java 向量计算函数 Java 适配函数的源代码，相关测试用例、测试报告及相关的函数说明文档: 《kml-java 向量计算函数 Java 接口说明书》 《kml-java 向量计算函数 Java 软件包测试报告》
阶段验收和总结	2021/06/30	合作项目验收汇报	完成所有功能函数的交付及验收，作总结汇报。	本期 kml-java 向量计算函数 Java 适配开发报告。

T: 合同生效日期

### 4. 项目监控和管理

本项目采用如下项目管理机制:

1. 以邮件方式项目周报。
2. 每双周电话会议交流项目进展。
3. 根据实际需要安排的其它交流，如电话、邮件、电话会议等。