

### 鲲鹏技术 应用软件迁移测试技术规范

KunpengTechnology specification for application software system migration testing

(征求意见稿)

XXXX - XX - XX 发布

XXXX - XX - XX 实施

## 前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件由辽宁省工业和信息化厅提出并归口。

本文件起草单位：北方实验室（沈阳）股份有限公司、辽宁鲲鹏生态创新中心有限公司、辽宁省标准化研究院。

本文件主要起草人：张健楠、袁洪朋、李海涛、白杨、丁琳、张麇、刘文志、梁爽、郝博、肖宇鹏、闫丽杰、赵兴彦、邱学思、叶松、韩燕妮。

本文件发布实施后，任何单位和个人如有问题和意见建议，均可以通过来电和来函等方式进行反馈，我们将及时答复并认真处理，根据实际情况依法进行评估及复审。

归口管理部门通讯地址：辽宁省工业和信息化厅，联系电话：

标准起草单位通讯地址：沈阳市浑南新区三义街6-1号21层，联系电话：024-83785843。

# 鲲鹏技术 应用软件迁移测试技术规范

## 1 范围

本文件规定了应用软件从传统计算底座向新一代信息技术的鲲鹏计算底座迁移适配规范和术语、定义。

本文件适用于独立软件开发商和迁移适配服务机构对应用软件迁移适配时进行参考。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 11457 信息技术 软件工程术语

GB/T 25000.51—2016 系统与软件工程 系统与软件产品质量要求与评价（SQuaRE）第51部分：就绪可用软件产品（RUSP）的质量要求和测试细则

## 3 术语和定义

GB/T 11457界定的以及下列术语和定义适用于本文件。

### 3.1

**精简指令集 reduced instruction set**

一种执行较少类型计算机指令的微处理器。

注1：精简指令集也称为RISC指令集。

注2：RISC处理器执行较少的计算机指令类型，寻址方式更简单，能够以更快的速度执行操作。

### 3.2

**复杂指令集 complex instruction set**

一种按顺序串行执行的计算机中央处理器的设计模式。

注3：复杂指令集也称为CISC指令集。

注4：每条指令中的各个操作也是按顺序串行执行。

### 3.3

**ARM 架构 the ARM architecture**

微处理器执行的计算机语言指令集，是一个32位精简指令集（RISC）处理器架构。

### 3.4

**X86 架构 the X86 architecture**

微处理器执行的计算机语言指令集，是复杂指令集（CISC）处理器架构。

### 3.5

**鲲鹏 kunpeng**

采用ARM架构指令集设计的一种中央处理器的名称。

### 3.6

**迁移适配 migration adaptation**

通过对应用软件进行代码的重新编译、部署环境的重新配置，将应用软件从基于X86架构的计算环境迁移到基于鲲鹏ARM架构的计算环境。

## 4 应用软件迁移过程

### 4.1 迁移流程

应用软件迁移包括信息收集、分析迁移、代码编译、软件包检查、测试验证五个步骤，见图1应用软件迁移适配流程图。

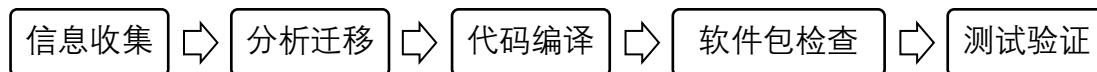


图1 应用软件迁移适配流程图

### 4.2 信息收集

#### 4.2.1 软件信息收集包括：

- a) 自研软件：收集自研软件开发语言类型、软件架构、环境配置等；
- b) 开源软件：收集开源软件版本信息；
- c) 商业软件：收集开源软件版本信息；
- d) 中间件：收集中间件品牌、型号、版本；
- e) 编译器：收集编译器品牌、型号、版本；
- f) 操作系统：收集操作系统品牌、型号、版本；
- g) 数据库：收集数据库品牌、型号、版本。

#### 4.2.2 硬件信息收集包括：

- a) 芯片：收集芯片型号、指令集架构；
- b) 整机：收集整机品牌、型号。

#### 4.2.3 信息收集内容如图 2 所示。

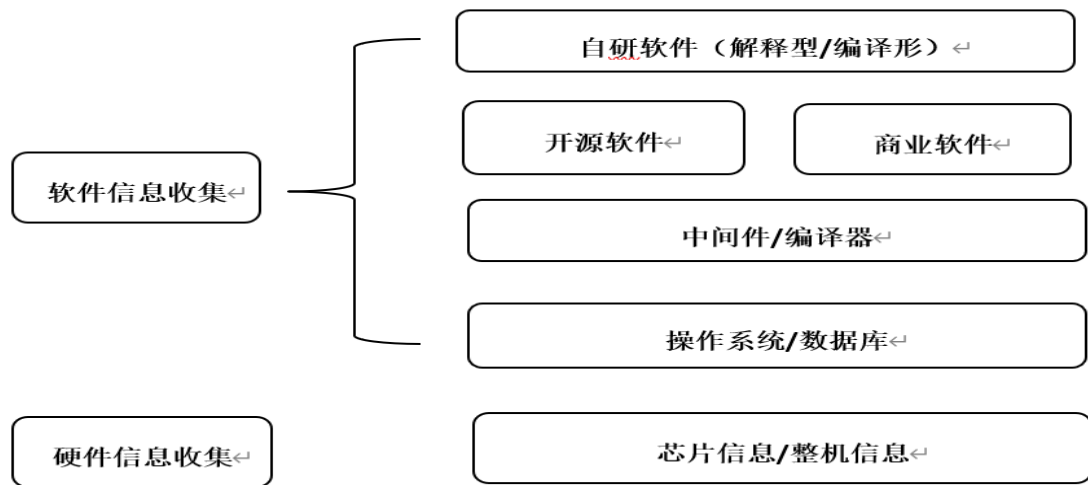


图2 信息收集内容

### 4.3 分析迁移

#### 4.3.1 总则

分析迁移共分为五部分，分为自研软件分析迁移、开源软件分析迁移、商用软件分析迁移、基础软件分析迁移和迁移环境分析。在此阶段需分析软件栈与鲲鹏ARM架构的兼容性，将不兼容的部分进行修改、替换。

#### 4.3.2 自研软件分析迁移

分析自研软件开发语言类型、软件架构、环境配置等是否支持鲲鹏ARM架构，自研软件代码编译详见第4.4章 代码编译部分内容。

#### 4.3.3 开源软件分析迁移

获取开源软件包，查看软件包版本是否兼容鲲鹏ARM架构。如不兼容，则获取开源软件ARM64软件包或下载源码重新编译。

#### 4.3.4 商用软件分析迁移

分析商用软件的版本是否兼容鲲鹏ARM架构。如不兼容，则获取商用软件ARM版本或更换其他可兼容的同类商用软件。

#### 4.3.5 基础软件分析迁移

分析迁移需考虑：

- 分析中间件品牌、型号、版本是否兼容鲲鹏 ARM 架构，如不兼容，则需替换兼容的中间件；
- 分析编译器品牌、型号、版本是否兼容鲲鹏 ARM 架构，如不兼容，则需替换兼容的编译器；
- 分析操作系统品牌、型号、版本是否兼容鲲鹏 ARM 架构，如不兼容，则需替换兼容的操作系统；
- 分析数据库品牌、型号、版本是否兼容鲲鹏 ARM 架构，如不兼容，则需替换兼容的数据库。

#### 4.3.6 迁移环境分析

分析迁移环境相关信息，如迁移使用的整机品牌型号、CPU型号和所使用的指令集架构。为实现迁移目的，需使用鲲鹏ARM架构CPU和兼容的服务器。

#### 4.4 代码编译

##### 4.4.1 源码编译

###### 4.4.1.1 编译型语言：

- a) 修改软件源码中对 X86 架构依赖的宏定义及函数；
- b) 替换支持鲲鹏 ARM 架构的编译选项；
- c) 将不兼容鲲鹏 ARM 架构的内联汇编函数重新编译；
- d) 将不兼容鲲鹏 ARM 架构的指令重新编写。

###### 4.4.1.2 解释型语言：

- a) 对软件中使用编译型语言开发的部分进行重新编译；
- b) 安装 ARM 架构版本的软件开发工具包或虚拟机。

##### 4.4.2 软件包迁移

###### 4.4.2.1 编译型语言：

编译型语言需重新编译依赖库，检查软件包的运行依赖库和可执行程序，按照软件安装包原有结构重新制作成鲲鹏的软件包。

###### 4.4.2.2 解释型语言：

解释型语言基于对应的虚拟机运行，按照软件安装包原有结构重新制作成新的软件包。

#### 4.5 软件包检查

将修改好的源码重新打包、检查，确认是否存在不兼容的依赖文件，将打包好的软件安装到服务器中。

#### 4.6 测试验证

##### 4.6.1 功能测试

对迁移适配后的软件功能进行测试，包括单元测试、系统集成测试。

##### 4.6.2 性能测试

4.6.2.1 对迁移适配后的软件业务性能进行测试。

4.6.2.2 监测 CPU、内存、网卡、磁盘各项硬件指标是否正常。

##### 4.6.3 长稳测试

长时间运行迁移适配后的软件，检测软件能否长期稳定运行。