



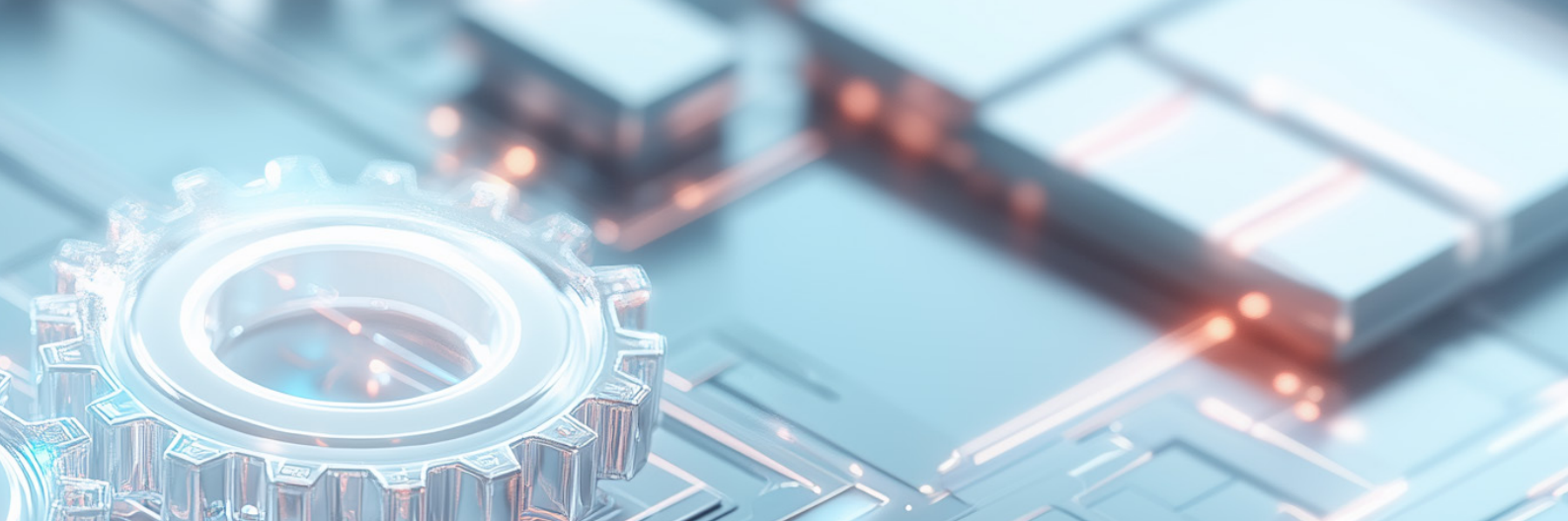
# 鲲鹏原生开发

技术白皮书



# 目录 CONTENTS

<b>01</b>	<b>鲲鹏原生开发的机遇和挑战</b>	<b>01</b>
<b>02</b>	<b>鲲鹏原生开发的核心技术理念</b>	<b>03</b>
<b>03</b>	<b>鲲鹏原生开发能力介绍</b>	<b>05</b>
	3.1 代码开发阶段	06
	3.1.1 代码开发	06
	3.1.2 代码优化	28
	3.1.3 编译	30
	3.1.4 调试	32
	3.1.5 调优	37
	3.2 流水线阶段	61
	3.2.1 门禁检查	61
	3.2.2 编译构建	66
	3.2.3 调优	67
	3.2.4 测试	68
	3.2.5 版本发布	69
	3.3 OS底座（openEuler系操作系统）	70
	3.3.1 openEuler简介	70
	3.3.2 openEuler基础能力	71
	3.3.3 openEuler工具和社区服务	75



# 04

## 行业案例

80

<b>4.1 金融</b>	81
4.1.1 恒生电子：基于鲲鹏原生开发经纪场外业务系统，让证券数据处理更高效	81
4.1.2 宇信科技：基于鲲鹏原生开发金融软件，让金融信息处理更高效	83
<b>4.2 电信</b>	86
4.2.1 浩瀚深度：鲲鹏原生赋能 DPI 采集系统，网络数据可视化处理更加高效	86
4.2.2 恒安嘉新：基于鲲鹏原生开发安全采集分析平台，保障企业信息系统安全稳定运行	89
<b>4.3 政府</b>	91
4.3.1 超图软件：基于鲲鹏原生开发地理信息系统，让自燃资源评估更高效	91
4.3.2 深圳防灾减灾技术研究院：鲲鹏原生助力地震观测数据高效处理	93
<b>4.4 电力</b>	95
4.4.1 广州海颐：基于鲲鹏原生开发高性能、高可靠电力系统软件，加速电网数字化转型	95
4.4.2 国能信控：基于鲲鹏原生开发新能源计算平台，为电力系统提供高效算力支撑	97
<b>4.5 医疗</b>	99
4.5.1 卫宁健康：基于鲲鹏原生开发新一代医疗信息系统WiNEX，让医疗服务更便利	99
4.5.2 东华软件：构建鲲鹏原生医疗经营管理系统，让医院运作更有序	101
<b>4.6 水平软件</b>	103
4.6.1 星环科技：基于鲲鹏原生大数据基础软件，让企业数据流转更快捷	103
4.6.2 深信服：联合鲲鹏共同推进企业级分布式存储EDS原生开发创新实践，让数据管理更可靠	105

01

鲲鹏原生开发的

# 机遇和挑战



鲲鹏计算产业聚焦计算架构创新和开源基础软件的研发，致力于推动鲲鹏生态发展。通过战略性、长期性的研发投入，吸纳全球计算产业的优秀人才和先进技术，持续推进全栈计算技术的创新发展，加快构筑面向多样化计算的全球开源体系与产业标准。基于“硬件开放、软件开源、使能伙伴、发展人才”的策略推动鲲鹏计算产业发展。

目前，鲲鹏软硬件联合创新已覆盖国计民生核心场景，主流伙伴核心应用已迁移到鲲鹏，鲲鹏生态进入快速发展期。当前阶段，鲲鹏和 x86 多平台版本并行迭代成为主要需求，由于两个架构开发生态的差异，如果采用两套流水线分别开发两套代码，则会带来很多重复开发验证的工作，并且两个版本分别开发、构建和维护效率低。因此，鲲鹏创新性的提出鲲鹏原生开发：基于鲲鹏硬件 + openEuler + 鲲鹏 DevKit + 鲲鹏 BoostKit，实现 1 套代码 + 1 条流水线 + 多平台版本，助力伙伴持续且首发性能领先的商用版本。



02

鲲鹏原生开发的

# 核心技术理念



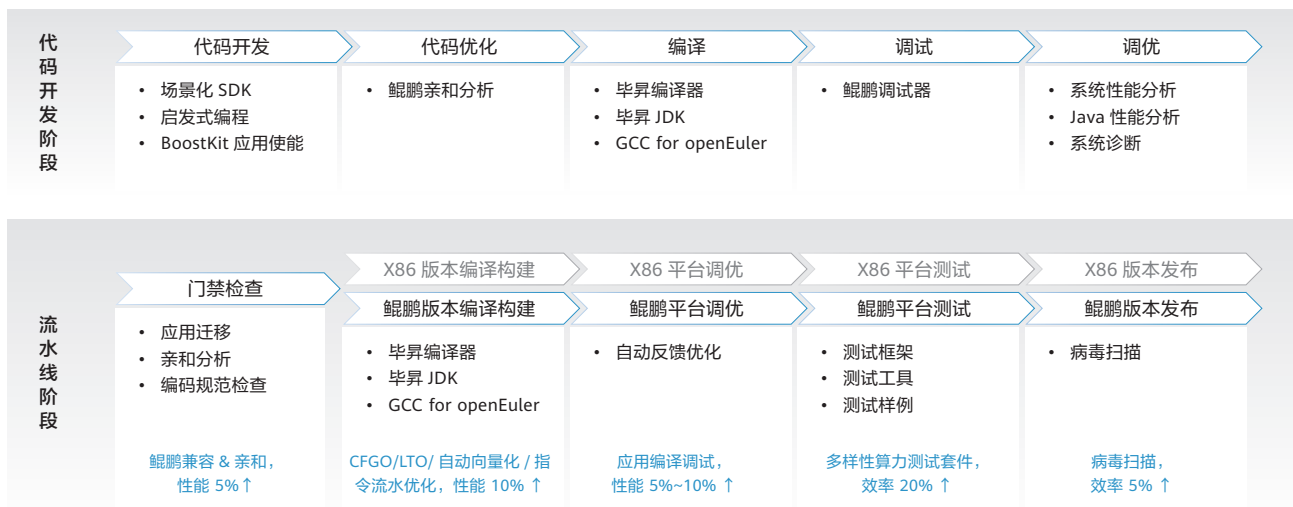
鲲鹏原生开发的核心技术理念：

基于鲲鹏硬件 +openEuler+ 鲲鹏开发套件 DevKit+ 鲲鹏应用使能套件 BoostKit，实现 1 套代码 +1 条流水线构建多平台版本，效率更高、性能更优。

鲲鹏原生开发包含代码开发阶段和流水线阶段：

- » 代码开发阶段：通过鲲鹏 DevKit、BoostKit 开发代码，充分应用鲲鹏架构优势，性能更优。
- » 流水线阶段：鲲鹏 DevKit 以命令行方式 1 小时接入 CI/CD，便捷发布多平台版本。

图 2-1 核心技术理念



- » 极简融入 & 高效发布：DevKit 原生开发插件 1 小时接入两大主流（Jenkins、GitLab）CI/CD 流水线，高效提升鲲鹏流水线搭建及版本发布效率。
- » 鲲鹏亲和 & 极致性能：鲲鹏流水线使能鲲鹏亲和分析工具、编译器、性能工具等，实现应用性能提升。
- » 多架构兼容：1 套流水线同时支持多样性算力版本构建及测试。

03

鲲鹏原生开发

# 能力介绍



# 3.1 代码开发阶段

## 3.1.1 代码开发

### 3.1.1.1 代码开发介绍

鲲鹏原生开发在代码开发阶段提供了 DevKit 应用开发工具和 BoostKit 应用使能套件，可通过启发式编程及多场景化 SDK 提供的示例 Demo 和文档，有效提升鲲鹏原生开发效率。

图 3-1 代码开发阶段

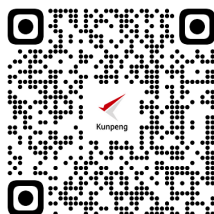


### 3.1.1.2 应用开发工具

应用开发工具支持创建鲲鹏应用工程，支持 C/C++ 开发语言，编码时能够自动匹配鲲鹏加速库函数字典、智能提示、高亮、联想字典中可以替换的库和函数。支持以下功能：

- » 鲲鹏应用工程：只需要在创建鲲鹏应用工程页面进行简单的输入和选择，便可以实现自动化构建鲲鹏应用工程，包括空工程、通用计算应用工程、安全计算应用工程、高性能计算应用工程、DPAK 应用工程和数据 IO 应用工程。
- » 字典管理：支持加速库函数字典管理，可线上（自动）和线下更新。
- » 编程辅助：支持鲲鹏加速库函数的悬浮提示、函数搜索，支持 Coding 时自动联想和高亮鲲鹏加速库优化后的相关函数。

应用开发工具，详细介绍与最新内容请参见[鲲鹏社区](#) → [鲲鹏开发套件 DevKit](#) → [开发](#)，详情可扫描下方二维码。

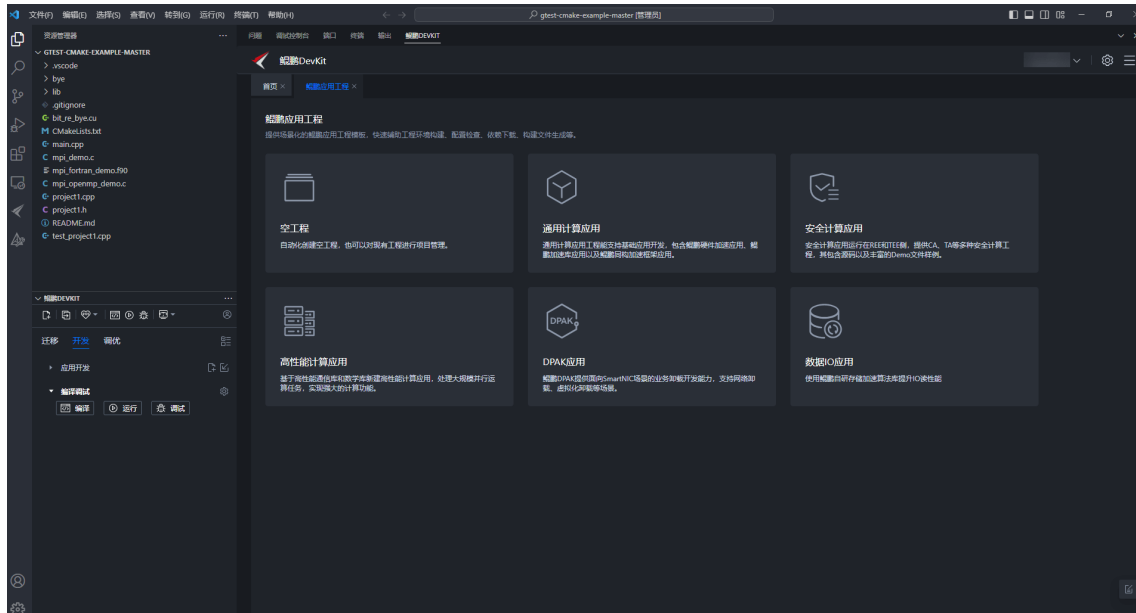


## 鲲鹏应用工程

提供场景化的鲲鹏应用工程模板，快速辅助工程环境构建、配置检查、依赖下载、构建文件生成等。

**步骤 1** 创建鲲鹏应用工程（通用计算应用工程，界面截图以 VS Code 环境为例）。

图 3-2 创建工程



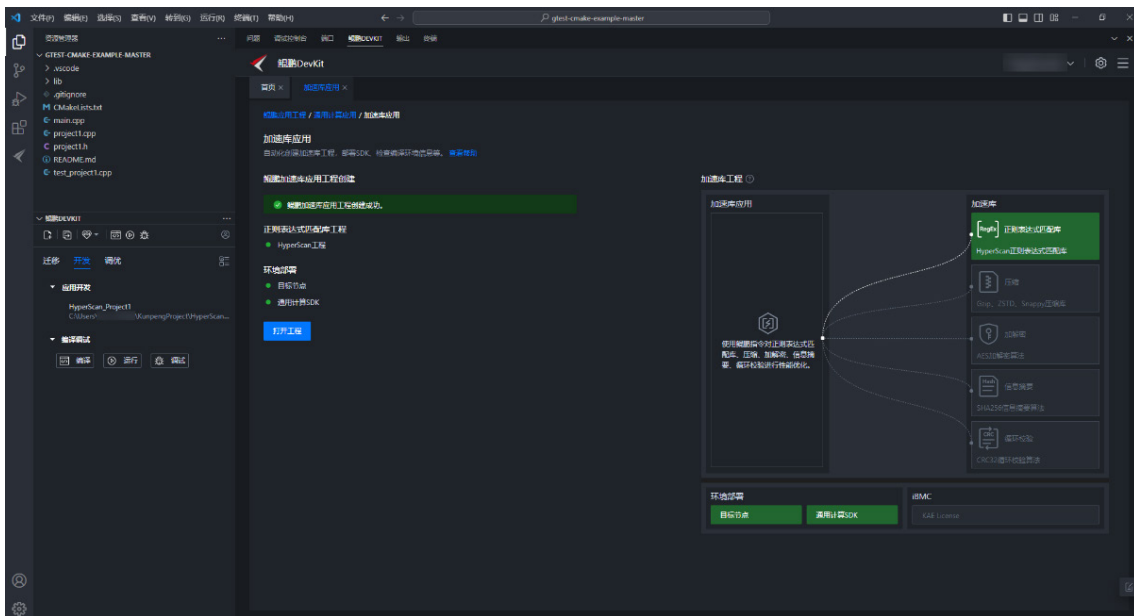
**步骤 2** 部署 SDK。

图 3-3 部署 SDK



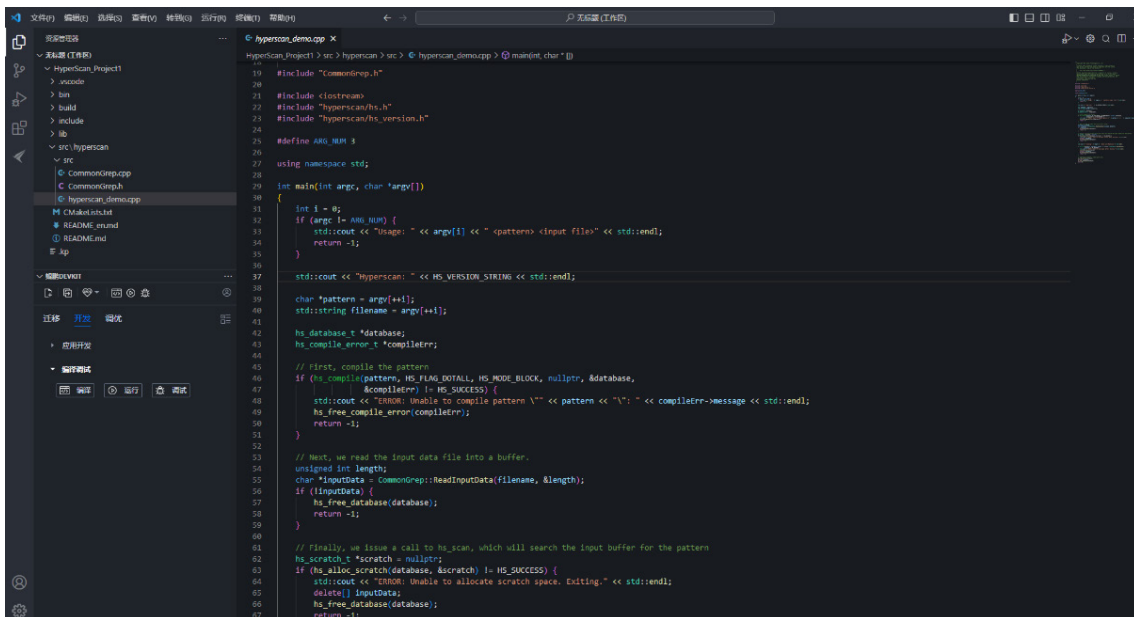
步骤 3 工程创建成功后，展示结果。

图 3-4 工程创建成功



步骤 4 单击“打开工程”，查看工程样例代码。

图 3-5 查看样例代码



---- 结束

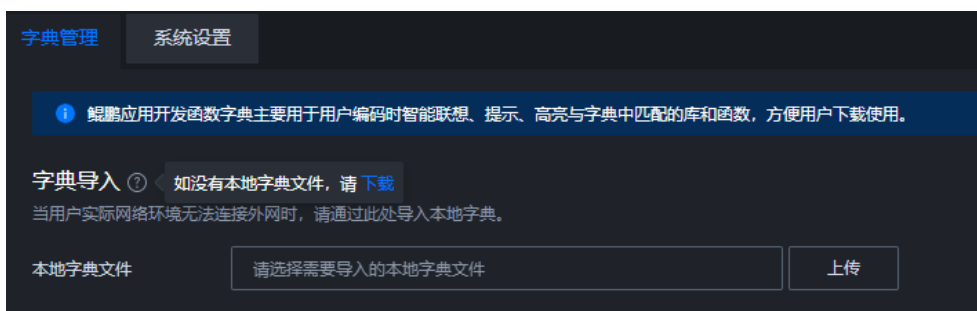
## 字典管理

支持加速库函数字典管理，可线上（自动）和线下更新。

**步骤 1** 当用户实际网络环境无法连接外网时，可上传本地字典文件。

鼠标移动到“字典导入”旁的 ，单击“下载”下载字典文件并在本地字典文件框内上传。

图 3-6 上传本地字典文件



**步骤 2** 开启加速库提示，开启后，用户编码过程中将会自动联想、补齐与字典中匹配的库和函数，方便用户下载使用。

图 3-7 开启加速库提示



**步骤 3** 开启编译选项智能联想，开启后，工具将会根据用户选择的编译器版本进行智能联想，可在编译器版本下拉列表中选择合适的编译器版本。

图 3-8 开启编译选项智能联想



### 说明

用户构建脚本时，工具将会根据用户选择的 GCC 版本或 BiSheng Compiler 版本进行智能联想，BiSheng Compiler 版本范围在 2.1.0~4.0.0，GCC 版本范围在 4.8.5~10.3.1。

**步骤 4** 设置跳过证书验证的网站，设置后的网站无需进行证书验证。单击“添加网站”，在弹框中输入网站地址。

图 3-9 添加网站

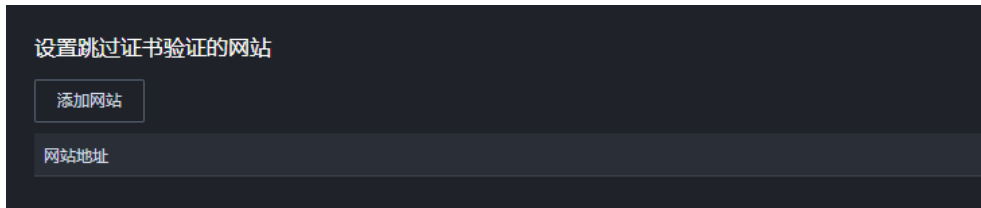


图 3-10 输入网站地址



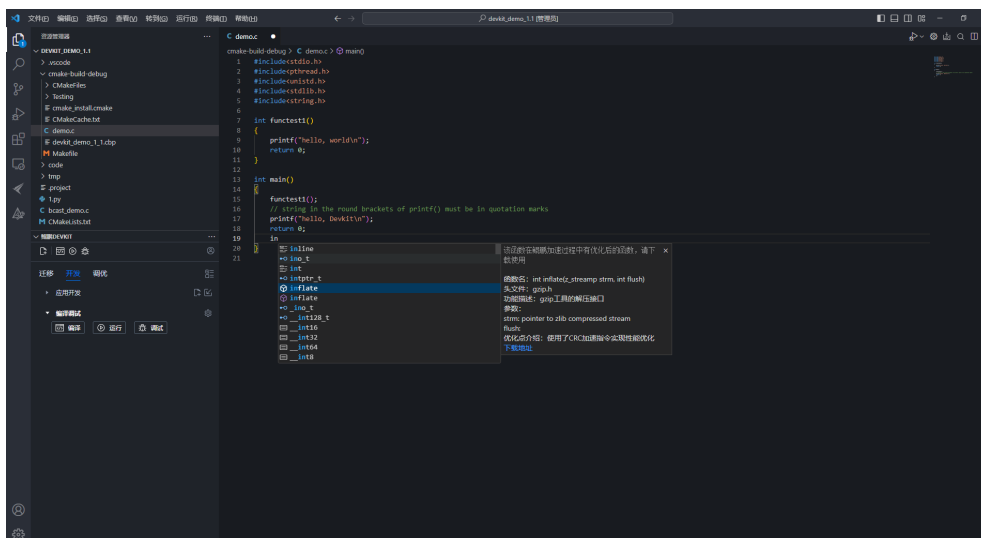
---- 结束

## 编程辅助

支持鲲鹏加速库函数的悬浮提示、函数搜索，支持编码时智能联想和高亮鲲鹏加速库优化后的相关函数。

**步骤 1** 在代码编辑区输入代码，工具会根据输入的内容自动联想函数或补全代码。

图 3-11 自动联想函数



步骤 2 通过代码编辑区右上角的  ( 放大镜 )，在搜索框输入关键字，可实现函数搜索。

图 3-12 函数搜索



---- 结束

### 🔗 3.1.1.3 鲲鹏 BoostKit 应用使能套件

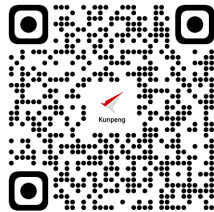
#### 1) 鲲鹏 BoostKit 应用使能套件简介

鲲鹏 BoostKit 应用使能套件，基于鲲鹏硬件、基础软件和应用软件的全栈优化，提供高性能开源组件、基础加速软件包和应用加速软件包，使能应用极致性能，其针对大数据、分布式存储、数据库、虚拟化和 ARM 原生等场景进行了深度优化鲲鹏架构特性，如内存管理、计算调度等方面的技术优势，通过预置的高性能库和框架，赋能开发者轻松构建适应鲲鹏架构的高性能应用。在同一个构建流程中，当集成 BoostKit 后，基于鲲鹏架构的目标软件包能够获益较大的性能提升。

图 3-13 鲲鹏 BoostKit 应用使能套件简介



获取最新的鲲鹏 BoostKit 应用使能套件和详细的用户指南，请参见[鲲鹏应用使能套件 BoostKit](#)，详情可扫描下方二维码。



## 2) 鲲鹏 BoostKit 应用加速

鲲鹏 BoostKit 提供性能倍增的应用加速软件包，使能数据处理极致性能、数据访问极致高效和云手机极致体验。

鲲鹏 BoostKit 应用加速软件包，详细介绍与最新内容请参见[鲲鹏应用使能套件 BoostKit](#) → [应用加速软件包](#)，详情可扫描右方二维码。



图 3-14 鲲鹏 BoostKit 场景化基础加速能力介绍

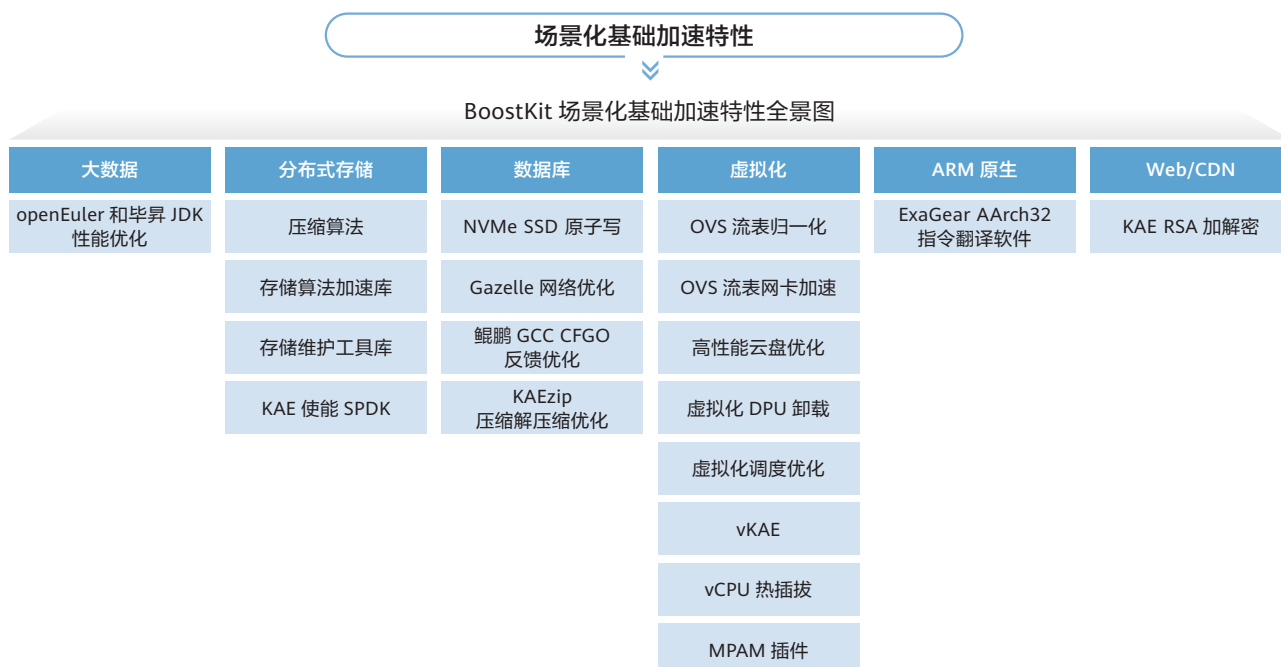


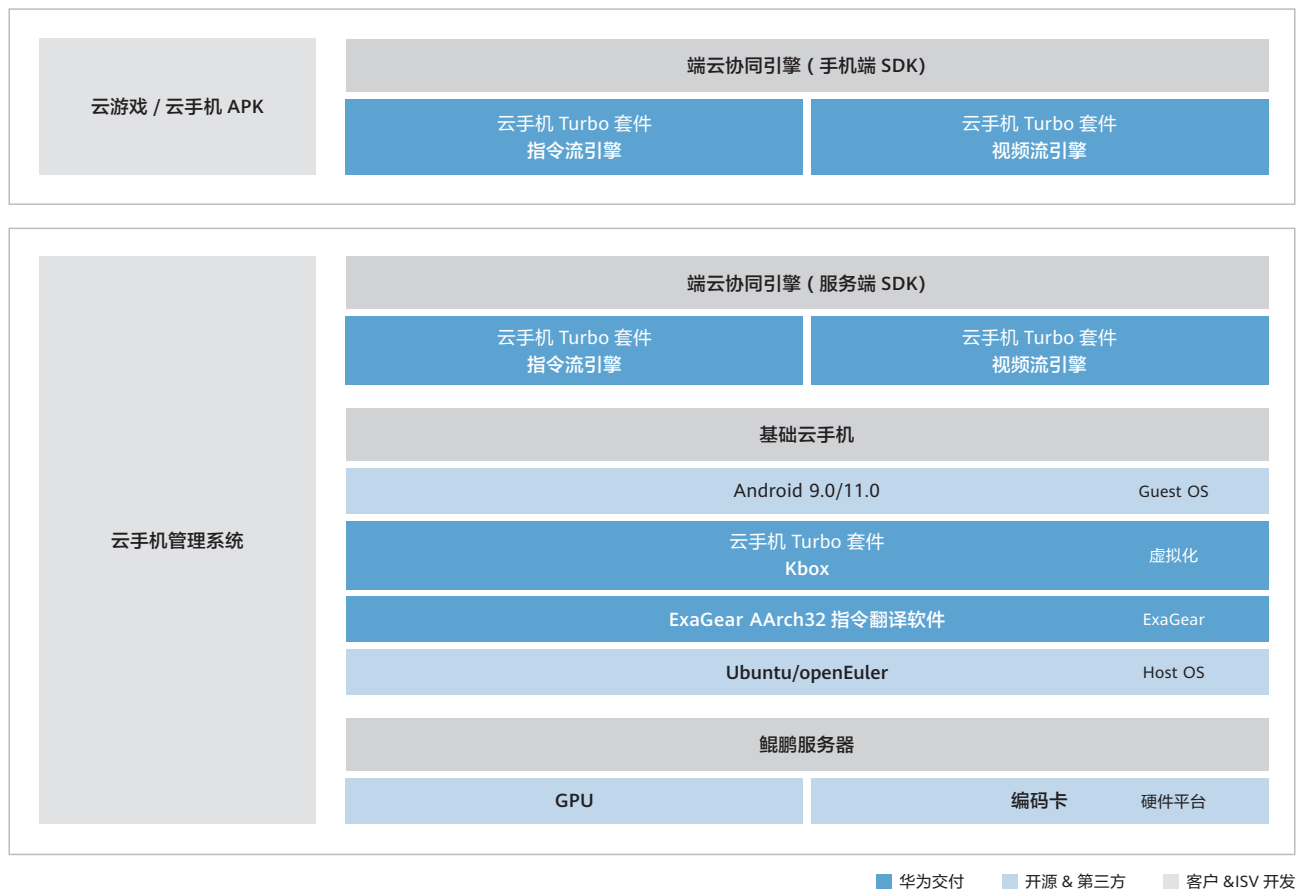
图 3-15 鲲鹏 BoostKit 场景化应用加速能力介绍



## ARM 原生

鲲鹏 BoostKit ARM 原生利用 ARM 指令集同构优势，支持移动应用无损上云，同时将多年技术积累浓缩到 Kbox 云手机容器、指令流引擎、视频流引擎核心能力等组件，形成了云手机 Turbo 套件，降低了开发难度，提升整机的密度，降低云手机单路成本，其中 HostOS 支持 Ubuntu 和 openEuler，GuestOS 支持 android-9.0.0\_r55 和 android-11.0.0\_r48，用户可以基于云手机 Turbo 套件进行二次开发，从而实现云手机极致的性能和业务体验。

图 3-16 鲲鹏 BoostKit ARM 原生云手机整体架构



### ExaGear AArch32 指令翻译软件

ExaGear AArch32 指令翻译软件为鲲鹏服务器提供 AArch32 特性，基于鲲鹏服务器在 ARM 原生场景下能够完全兼容 AArch32 的应用，保障 AArch32 V8.0 指令应用 100% 兼容。同时也支持 pre-translator 特性，优化翻译后的 APP 启动时间，提升客户体验。

### Kbox 云手机容器

Kbox 云手机容器提供了软件定义手机的基础能力，其基于鲲鹏服务器，在 Docker 容器技术和 Android 开源项目（AOSP）基础上，实现了将 GPU 设备直通到容器的轻量级设备仿真层架构方案，提供了基于 Android 系统的云手机容器参考方案。可实现在鲲鹏服务器支持 100 路 720p@30fps 的 Kbox 云手机容器高并发（以托管场景为例，云手机并发密度取决于客户云手机实际应用），具有高密度、高兼容性等商业价值。

### 视频流引擎

视频流云手机方案基于视频流的端云协同引擎，提供了低时延的云手机画面同步能力，支持 H.264 和 H.265 硬编码，在画质相当的情况下，可实现 H.265 编码带宽减少 30%+，其实现原理是利用云端强大的能力将应用和游戏的运行、渲染成最终的画面，并进行视频的压缩流化后，发送到终端播放显示。视频流引擎支持视频编码能力、视频解码播放能力、云手机图像截屏能力、触控和音频抓取 / 播放能力等核心功能，客户可以基于这些引擎进行二次开发，实现在移动终端上进行操控应用和游戏等操作。云端采用专业显卡进行渲染，可以获取高画质的业务体验；对终端的要求极低，只要求有视频解码能力即可；提供统一 API，降低二次开发难度，易集成。

### 指令流引擎

指令流云手机方案采用业界独创的端云分离渲染技术，可实现云侧免 GPU 部署，整机硬件成本下降 10%，其实现原理是云手机利用云端强大的算力，通过引擎实现云端复制应用和游戏的渲染指令，并进行渲染指令和纹理数据压缩流化，在端侧使用手机终端的 GPU 把这些指令渲染出来图像。指令流引擎，支持指令流分离渲染、纹理数据视频流化、触控和音频抓取 / 放能力等核心功能，客户可以基于这些引擎进行二次开发，实现在移动终端上进行操控应用和游戏等操作。通过指令流引擎技术，可以支持云手机全系统渲染，并提供近乎无损的画质，在 1080P/2k/4k 分辨率下均不影响传输带宽，并通过资源缓存技术，有效降低网络带宽 50%+；突破云端 GPU 能力限制，实现无 GPU 高密运行机制，单路硬件成本降低 40%，支持图形渲染状态机的本地执行与远端同步，实现 1080P 30FPS 低响应时延的用户体验。

## 大数据

鲲鹏 BoostKit 大数据聚焦大数据查询效率低、性能优化难等挑战，提供大数据组件的开源使能和调优、OmniRuntime、机器学习和图分析算法加速库等应用加速软件包，提升大数据分析效率。

图 3-17 大数据应用加速软件包介绍

大数据加速底座 OmniRuntime			算法加速库
OmniData 算子下推	OmniOperator 算子加速	OmniShuffle Shuffle 加速	机器学习算法加速库
OmniMV 物化视图	OmniAdvisor 参数调优	OmniHBaseGSI 全局二级索引	图算法加速库

### openEuler 和毕昇 JDK 性能优化

适用 Hive 2.X/3.X、Spark 2.X，openEuler 操作系统基于大数据核心组件 Hive、Spark 通过磁盘 IO、网络 IO 的调度策略优化、Neon 指令优化等实现大数据计算性能提升。毕昇 JDK 性能优化基于鲲鹏处理器构建开源 JDK 社区，通过 AppCDS、GC 算法优化、编译优化等提升大数据核心组件 Hive、Spark 的计算性能。Hive 性能提升 2%~25%，Spark 性能提升 3%~25%。

### OmniRuntime

大数据 OmniRuntime 是鲲鹏 BoostKit 大数据面向应用加速推出的一系列特性，包括 OmniData 算子下推、OmniOperator 算子加速、OmniShuffle Shuffle 加速、OmniMV 物化视图、OmniAdvisor 参数调优和 OmniHBaseGSI 全局二级索引，旨在通过插件化的形式，端到端提升数据加载、数据计算和数据交换的性能，从

而提升大数据分析的性能。Spark 使用 OmniRuntime 加速特性执行 SQL 计算，相比原生性能提升 20%~40%，具体包括组件如下：

#### » OmniData 算子下推

适用于存算分离场景或大规模存算融合场景，支持 Spark 3.0.0/3.1.1、Hive 3.1.0 ( Tez 0.10.0 )，是一种将大数据引擎的算子下推到存储节点或卸载节点的服务，从而实现了近数据计算，减少了网络带宽，将该特性集成到 Spark 后，基于 TPC-H 测试用例 12 条算子下推的 SQL 性能平均提升 40%。集成到 Hive 后，基于 TPC-H 测试用例 4 条算子下推 SQL 性能平均提升 20%。

#### » OmniOperator 算子加速

适用于虚拟化场景，支持 Spark 3.1.1、Spark 3.3.1、Hive 3.1.0 版本，其采用 Native Code ( C/C++ ) 实现大数据 SQL 算子来提高查询性能的特性，通过列式存储和向量化执行技术，同时利用鲲鹏加速库，提升算子的执行效率，将该特性集成到 Spark 后，基于 TPC-DS 99 条 SQL 验证，可实现 Spark 性能提升 30%。

#### » OmniShuffle Shuffle 加速

适用于虚拟化场景，支持 RDMA 和 TCP 两种网络模式，支持 Spark 3.1.1、Spark 3.3.1、Hive 3.1.0 版本，其基于 TCP/RDMA 等网络介质，优化数据分析过程中跨节点的数据写入、传输和读取流程，提升 Shuffle 性能，支持数据分析过程性能提升，将该特性及算子加速特性集成到 Spark 后，基于 TPC-DS 99 条 SQL 验证，可实现 Spark 性能提升 40%。

#### » OmniMV 物化视图

适用于虚拟化场景，支持 Spark 3.1.1 版本，支持 ClickHouse 22.3.6.5 版本，其通过 AI 算法从历史 SQL 查询中推荐出最优物化视图，并在 Spark 中自动对用户 SQL 进行物化视图匹配，将匹配成功的物化视图替换用户执行计划的部分 SQL，大幅减少重复计算，提升查询性能，在将该特性集成到 Spark 后，基于 TPC-DS 基准测试用例实现了 Spark 组件计算性能平均提升 30%，集成到 ClickHouse 后，基于 Star Schema Benchmark 实现 ClickHouse 计算性能平均提升数倍。

#### » OmniAdvisor 参数调优

适用于虚拟机场景，支持 Spark 3.1.1 和 Hive 3.1.0 ( 只支持 Hive on Tez 模式 )，因 Spark/Hive 引擎参数众多，取值范围大，人工调优存在调优效率低，调优效果不佳，OmniAdvisor 旨在通过 AI 的方式，实现参数的自动推荐，从而提升调优效率和调优效果。集成该特性后，基于 TPC-DS 10 条 SQL 可实现 Spark 性能提升 10%。

#### » OmniHBaseGSI 全局二级索引

适用于虚拟机场景，支持 HBase 2.4.14 版本，因 HBase 原生提供了主键索引，但若使用非 Rowkey 进行查询，则需要全表扫描，不仅耗费大量资源，查询时延也很长，OmniHBaseGSI 全局二级索引，可以在非 Rowkey 列上创建全局二级索引，从而极大加速非 Rowkey 列的查询性能，集成该特性后，可实现在 100 并发下，平均时延小于 30ms，P99 时延小于 300ms。

### 机器学习和图分析算法

兼容 Spark 原生机器学习和图分析算法的 API 接口，支持适配 Spark 2.3.2、Spark 2.4.5、Spark 2.4.6 版本，部分算法支持 Spark 3.1.1 和 Spark 3.3.1 版本，基于算法原理和芯片特征针对机器学习和图分析算法进行深入优化，从而可以大幅提升大数据算法场景的计算性能。鲲鹏的机器学习 & 图分析算法加速库相比业界 Spark 原生 MLlib 和 GraphX，相同精度下计算性能提升 20% 以上。

## 分布式存储

鲲鹏 BoostKit 分布式存储使能套件聚焦开源 Ceph 存储的性能低、成本高等关键挑战，通过存储算法加速库和存储 Ceph 加速库等特性提升系统性能和降低存储成本，充分发挥鲲鹏算力优势，提供高性价比存储方案。

### 压缩算法

适用于块存储服务数据压缩和对象存储服务数据压缩，与主流开源压缩算法相比，压缩率可提升 25%，带宽性能提升 10%。

### 存储算法加速库

支持 Ceph 14.2.8 版本，其采用鲲鹏优化的算法代替主流开源算法，提升存储性能。当前包括 EC 算法、CRC16 T10DIF 算法和 CRC32C 算法，具体介绍如下：

#### » EC 算法

适用于块存储或对象存储场景，支持 K+1、K+2、K+3、K+4 ( $2 \leq K \leq 25$ ) 和 28+3 配比，其他配比暂不支持，其采用向量化 EC 编解码方案，以低阶二元 XOR 操作替代传统标量编码的高阶有限域乘法，配合编码调度，复用中间计算结果减少操作数，与主流开源算法相比，EC 算法编码性能提升 1 倍以上。

#### » CRC16 T10DIF 算法和 CRC32C 算法

通过大数求余算法和配合鲲鹏向量化指令实现编码加速，与主流开源算法相比，CRC16 T10DIF 算法 4K 性能提升 100%，CRC32C 算法 4K 性能提升 20%。

### 存储维护工具库

存储维护工具库（KSML）是华为自研的存储维护工具库，包括 HDD/SSD 故障预测与 HDD/SSD 慢盘检测功能，基于机器学习算法，通过收集 SMART 数据训练模型，预测与识别存储集群潜在故障盘，通过采集系统磁盘的 svctm 完成慢盘检测。



## KAE 使能 SPDK

SPDK 的 BDEV 设备作为虚拟设备层对接底层多种设备类型（虚拟设备、物理设备），通过在 BDEV 设备中使能压缩和解密能够支持所有 SPDK 设备。鲲鹏 KAE 通过 Zlib 和 Openssl 提供压缩和解密，通过在 BDEV 设备中支持 Zlib 和 OpenSSL 的 KAE 加速实现对应能力的硬件卸载。

## EC Turbo

支持 Ceph 14.2.8 版本，适用于使用块存储或对象存储服务场景，不支持 Bcache，该特性是针对开源 Ceph 的 EC 流程进行优化，降低了数据读写流程中的 IO 放大比例，从而使得整体性能更高。相对于 Ceph 开源 EC，在均衡型配置下，对于块存储服务，EC Turbo 性能可达到 x86 三副本 80% 以上，存储成本降低 50%；对于对象存储服务，ECTurbo（4+2）性能达到 x86 三副本 80% 以上，大 IO 存储成本比三副本降低 50%，小 IO 成本与三副本持平。

## 智能写 Cache

支持 Ceph 14.2.8 版本，适用于块存储和对象存储服务写场景，其支持通过 IO 直通、QoS 控制策略、Writeback 控制策略以及 GC 控制策略，提升 Bcache 场景下的 Ceph 集群写性能。在块存储随机写场景下，IOPS 性能可提升 20% 以上。

## IO 直通

IO 直通工具是针对 Ceph 均衡型场景下的一个流程优化工具，可以自动对 Ceph 集群进行性能优化。在均衡型配置场景下，使用 IO 直通特性可提升存储性能 15% 以上。

## 数据压紧

通过消除补零对齐操作带来的数据浪费问题，结合压紧封装、空间计数分配、粒度分流、聚合提交、批量回调等手段提升数据缩减率并提升系统整体 IOPS，实现成本性能双收益。数据压紧可将数据压缩率再提高 20% 以上，对系统性能无损失。

## 元数据加速

元数据加速特性在 RocksDB 的基础上，结合华为自研算法进行了性能加速优化，在使能鲲鹏加速特性时可以获取更佳的性能。较开源 RocksDB，混合读写场景性能提升超过 30%。

## Ucache 智能读缓存

智能读缓存通过 IO 智能预取精准识别热点请求并针对顺序、间隔等 IO 流进行 IO 预取，将 IO 提前载入读缓存，同时读缓存通过 LRU 算法淘汰冷数据，从而提高缓存的 IO 命中率，提升读性能。Ucache 智能读缓存可提高读请求 IO 命中率以提升读性能。热点、顺序、间隔等 IO 流下性能提升 100%。

## BoostIO

在存算分离架构下，BoostIO 利用计算侧的内存和磁盘资源构建分布式多级缓存，写缓存通过 RDMA 高速通信、缓存亲和策略、副本冗余和线性布局等特性提升业务写性能，提高数据可靠性；读缓存通过数据预取提前将热点数据加载缓存磁盘中，通过 LRU 淘汰策略和冷热识别提高读缓存命中率，从而提升业务读性能。

## RDMA 网络加速

通过在 Ceph 网络框架 AsyncMessage 中新增插件支持 UCX 网络框架，实现 Ceph 全闪存场景支持网络全 RDMA 化。UCX 通信处理层主要包含 ceph 与 UCX 接口适配，并根据 RNDV 协议的特点，实现了零拷贝，提升大块写的性能。

## 机密计算

鲲鹏 BoostKit 机密计算 TrustZone 套件是基于 ARM TrustZone 技术的一个机密计算软件套件，包含华为自研 TEE（Trusted Execution Environment，可信执行环境）安全操作系统，鲲鹏服务器 BMC 和 BIOS 等，结合开源的操作系统驱动以及 SDK，旨在帮助伙伴更便捷地为行业客户构建机密计算解决方案，从而为用户的关键数据提供完整性、机密性保护和可信使用。

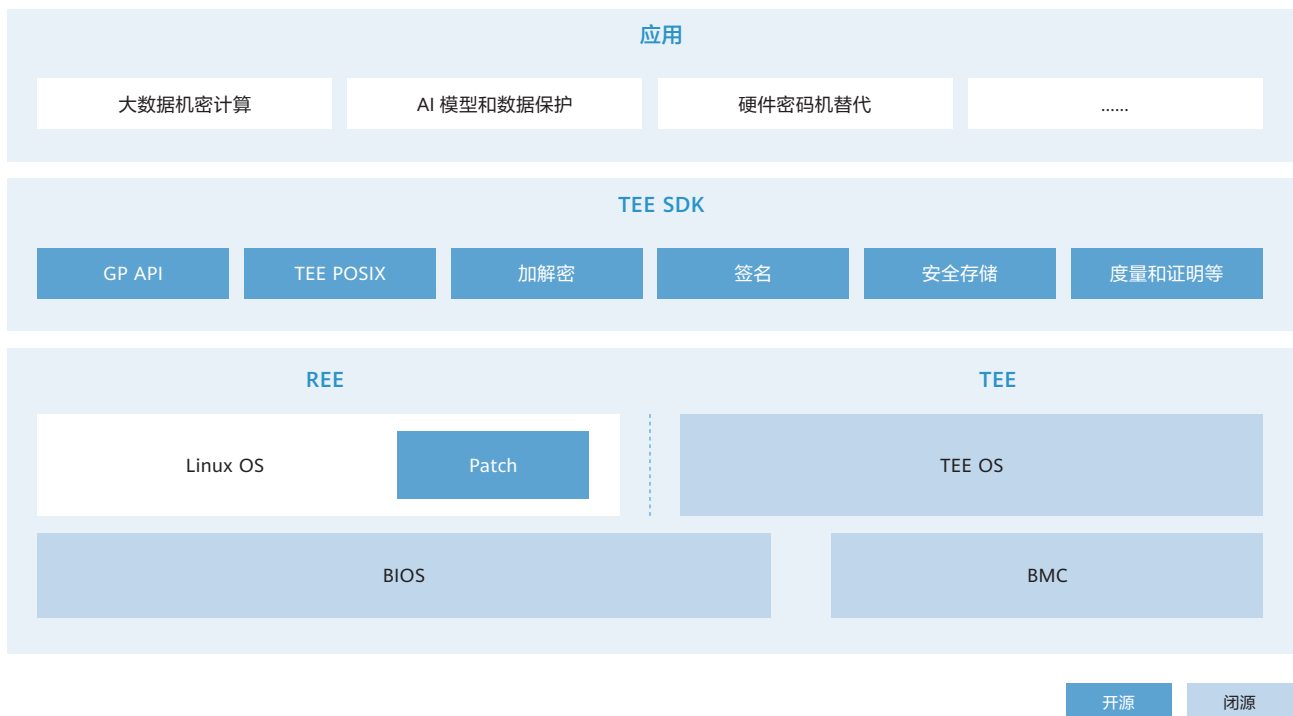
普通鲲鹏服务器默认不带有机密计算 TrustZone 完整套件，需要在购买鲲鹏服务器时明确带有 TEE 功能，支持 TEE 功能的服务器会在出厂阶段完成 TrustZone 套件预置安装。

iTrustee 基于 TrustZone 技术实现了整套安全解决方案，包含正常模式的客户端应用（Client Application，CA）、安全模式的可信应用（Trusted Application，TA）、安全模式下的可信操作系统。

iTrustee 应用于金融大数据数据挖掘场景，可保证数据处理过程中的机密性。可应用于一体化大数据中心场景，确保数据可信交易，同样也可用于行业隐私计算认证场景，确保计算过程中避免泄漏个人隐私信息。

iTrustee 安全可靠，其基于华为自研的微内核实现，安全 OS 已在手机侧商用近 10 年，用户数已过亿，同时安全性获得 CC EAL4+ 认证，兼容性获得 GlobalPlatform 认证。在规格方面也较为灵活，其中 TEE 侧安全内存支持按需配置，最大可配 512GB，可支持大数据、AI 等大型应用运行。

图 3-18 机密计算



- » TEE SDK：提供机密计算 REE（Rich Execution Environment，富执行环境）侧和 TEE 的接口、TA/CA 加密和签名工具、参考代码和接口说明等，方便用户快速构建应用。
- » REE Patch：操作系统的驱动，包括内核模块以及用户接口库。
- » TEE OS：华为自研安全操作系统，为可信应用提供加解密、安全存储等服务，并确保 TA 的完整性和机密性。

- » BIOS: 完成对 TEE OS 的解密以及验证, 确保 TEE OS 的机密性以及完整性。
- » BMC: 支持对 TEE OS 的升级维护。

## 数据库

鲲鹏 BoostKit 数据库对开源 MySQL OLAP 查询效率低、OLTP 场景高并发下锁导致的性能问题等关键挑战, 提供 MySQL 可插拔向量化分析引擎、MySQL 无锁优化、MySQL 可插拔线程池和 CRC32 指令优化等加速软件包, 深度优化了 OLAP 查询分析效率和 OLTP 在线交易事务处理能力, 充分发挥多核算力极致性能。提供主流开源和商业数据库最佳实践, 帮助开发者高效完成开源组件迁移和调优。

### NVMe SSD 原子写

适用 MySQL 各版本, 通过 SSD 硬件原子写特性消除 Doublewrite 双写软件冗余提升数据库的性能, 通过 SSD 硬件原子写特性密集写场景性能预计提升 15%。

### Gazelle 网络优化

适用 MySQL 各版本, 基于 DPDK 在用户态直接读写网卡报文, 共享大页内存传递报文, 使用轻量级 LwIP 协议栈。能够大幅提高应用的网络 IO 吞吐能力, 通过 Gazelle 网络优化, TPC-C 综合性能预计提升 10%。

### 鲲鹏 GCC CFGO 反馈优化

适用 MySQL 各版本, 采用多模态(源代码、汇编码、二进制)、全生命周期(编译、链接、后链接)的持续优化手段, 获取性能更优的目标程序。使数据库 TPC-C 综合性能提升 10%。

### KAEzip 压缩解压缩优化

支持使用 zlib 进行压缩和解压的 Greenplum 版本, 使用鲲鹏硬加速模块实现压缩、解压缩算法, 结合无损用户态驱动框架从而提升查询性能。采用 KAEzip 可以在到达硬件瓶颈之前, 在同一时间只处理一个请求、IO 占比多的场景下, 端到端的性能提升 10%。

### MySQL 并行查询优化

支持 MySQL 8.0.20、MySQL 8.0.25 版本, MySQL 单 SQL 查询只能调度单线程, 多核 CPU 无法使用, 单查询性能查询差难于满足查询场景的性能要求, 通过并行查询优化提升查询性能。可实现查询性能提升 >1 倍(性能提升与并行度有关)。

### MySQL 无锁优化

支持 MySQL 8.0.20 版本, 在 MySQL OLTP 场景下 DML 语句(Insert、Update、Delete)大量并发操作 trx\_sys 全局结构体中的关键数据结构, 造成临界区的竞争和同步瓶颈。MySQL 无锁优化改造后使用无锁哈希表维护事务单元, 减少锁冲突, 提升并发度, 可实现 Sysbench 写场景下性能提升 20%。

### MySQL 细粒度锁优化

支持 MySQL 8.0.20 版本, 在 MySQL OLTP 场景下 DML 语句(Insert, Update, Delete)大量并发操作访问 lock\_sys->mutex 全局锁保护的关键数据结构, 造成锁竞争严重导致性能下降。替换成细粒度 hash 桶锁。减少锁冲突, 提升并发度。可实现 TPC-C 综合性能预计提升 10%。

### MySQL NUMA 调度优化

支持 MySQL 8.0.20、MySQL 8.0.25 版本，在 MySQL OLTP 场景下高并发下系统默认的线程调度使得线程频繁跨 NUMA 的访问，这种情况导致 CPU 开销增大，性能提升受限制，需要对用户处理线程做动态绑定固定 NUMA CPU 减少跨 NUMA 访问，同时需要保证 CPU 访问的负载必须均衡，后台线程静态绑定固定 NUMA CPU 减少跨 NUMA 访问，提升后台线程效率。可实现 OLTP 场景性能提升 10%。

### MySQL 可插拔线程池

支持 MySQL 5.7.27、8.0.20、8.0.25、8.0.30 和 8.0.35 版本，仅基于 MySQL 8.0.25、8.0.30 和 8.0.35 的线程池特性支持可插拔动态加载。在 MySQL OLTP 场景下，高并发下线程数过多，CPU 消耗在无效的资源竞争和频繁切换上，线程池方案通过队列方式管理任务，所有的任务先放入等待执行队列，按系统执行能力取出任务队列让 CPU 执行，每个 CPU 同时处理任务个数是有限的，一般 2~5 个最优，从而保持稳定的业务处理能力。可实现 OLTP TPC-C 场景性能 10000 并发性能下降到最优的 10% 左右，开启线程池功能，性能可维持在 85%。

### CRC32 指令优化

提供支持 MySQL 8.0.25 版本的补丁包，该特性采用鲲鹏 CRC32 硬件指令替换 CRC32 算法的软件实现，从而提高系统业务的性能。通过 CRC32 指令优化特性，MySQL Sysbench 写场景性能有 5% 的提升。

### MySQL 可插拔在线向量化分析引擎

支持 MySQL 8.0.25 版本，该特性是 MySQL 预留接口第二执行引擎（Secondary Engine）的一种轻量实现，通过执行计划的并行计算，充分发挥鲲鹏 CPU 多核的优势，使 OLAP 性能倍级提升，且具有可插拔性，支持动态加载。采用并行加速技术，可将 OLAP 查询性能提升到 3 倍以上。



## 虚拟化

鲲鹏 BoostKit 虚拟化使能套件聚焦虚拟化轻载性能低、网络损耗大、资源碎片严重及开源生态可用性等关键痛点，提供了 OVS 流表网卡加速等特性提升系统性能，充分发挥鲲鹏多核架构、核间完全隔离的优势，释放鲲鹏极致算力。

### 负载感知加速系统

负载感知加速系统(以下简称 WAAS)能够基于每个计算任务深度调优,启用最优的加速库,自动配置全栈最佳参数。通过收集应用负载信息自动感知业务的应用类型、任务特征、OS 特征和微架构特征,对各层性能特征进行分析,生成调优策略,通过加速库加速、任务调优、OS 调优和 CPU 微架构调优对 SQL 任务进行全栈深度优化,动态调整达到业务实时最佳。

### OVS 流表归一化

在 OVS+DPDK 的基础上,提出流表归一化的方案,进一步加速云计算中的数据包转发性能,其典型场景为 VXLAN+CT 组网。卸载场景较不卸载场景网络转发性能提升 30% 以上。

### OVS 流表网卡加速

在虚拟化场景下,将 OVS 转发流表卸载到网卡硬件上,利用硬件的查表能力来提升流表的查找速度,提高虚拟化网络的处理能力,可实现虚拟化网络的转发性能提升 10 倍。

### 高性能云盘优化

通过 BoostKit 提供的 SPDK+Ceph 的方案来对虚拟化场景下的 Ceph 存储客户端进行加速,以提升存储场景的 IO 读写能力。存储 IOPS 性能提升 30% 以上。

### 虚拟化 DPU 卸载

虚拟化 DPU 卸载支持虚拟化场景下网络和存储的加速,其可以将本来运行在物理机上的软件(如:OVS-DPDK、SPDK 等)卸载到 DPU 卡上运行,节省了物理机的 CPU 负载,从而提高虚拟机密度,同时 DPU 卡支持 VirtIO-net 和 VirtIO-blk 等协议,其作为 VirtIO 设备后端,可提高虚拟化网络和存储性能。

### 虚拟化调度优化:

鲲鹏虚拟化通过软硬协同方案,加速虚拟机中应用对 CPU 的调度效率。

- » 通过 NUMA 感知和 cluster 感知特性,将关于 CPU 的拓扑结构直通到虚拟机,虚拟机 OS 内核可通过 cluster 任务调度优化选项,加速多线程进程调用效率。
- » 优化了抢占过程中的锁机制,提升虚拟机在超分场景下的性能。
- » 增加硬件死锁的机制,有效防止硬件死锁导致的虚拟机卡住而无法恢复的情况。

### vKAE:

鲲鹏加速引擎 KAE (Kunpeng Accelerator Engine) 是基于鲲鹏处理器提供的硬件加速解决方案,包含了 KAE 加解密和 KAEzip。vKAE 也可以在虚拟机或者容器中使用 KAE 能力。

### vCPU 热插拔:

虚拟机 vCPU 依照 ACPI 规范，模拟 ACPI GED 设备。在 vCPU 调整时，通过中断和处理函数方式，动态模拟 CPU 上下电。

### MPAM 插件:

通过限制离线业务对内存带宽和 L3 缓存容量的占用，避免离线业务干扰实时业务的性能:

- » 每个计算节点上部署 MPAM 插件，YAML 文件中配置资源组，每个资源组可指定内存带宽和 L3 缓存容量。
- » 部署离线业务时，在 YAML 文件中指定该业务归属的资源组。
- » MPAM 插件侦听到部署任务后，将容器中业务的进程 ID 配置到对应的资源组中（限制信息会通过 OS 配置到硬件芯片上）。

#### 说明

MPAM 涉及的共享资源包括：L2 Cache、L3 Cache 和 DMC 带宽。

## Web/CDN

鲲鹏 BoostKit CDN 使能套件聚焦 CDN 开源组件可用性和 CDN 缓存节点吞吐量低、时延大等问题，通过使能鲲鹏处理器内置的 RSA 加速引擎，对 RSA2048 算法进行硬件卸载，同时提供了 NUMA 优化等手段，以便充分发挥鲲鹏处理器多核优势，助力 CDN 缓存节点提供更大吞吐量，实现更低时延。

鲲鹏 BoostKit Web 使能套件聚焦 Web 应用 HTTPS 连接性能和 Web 开源组件可用性等关键问题，通过使能鲲鹏处理器内置的 RSA 加速引擎，提升 Web 网站的安全性，并实现用户 HTTPS 访问的极致体验。

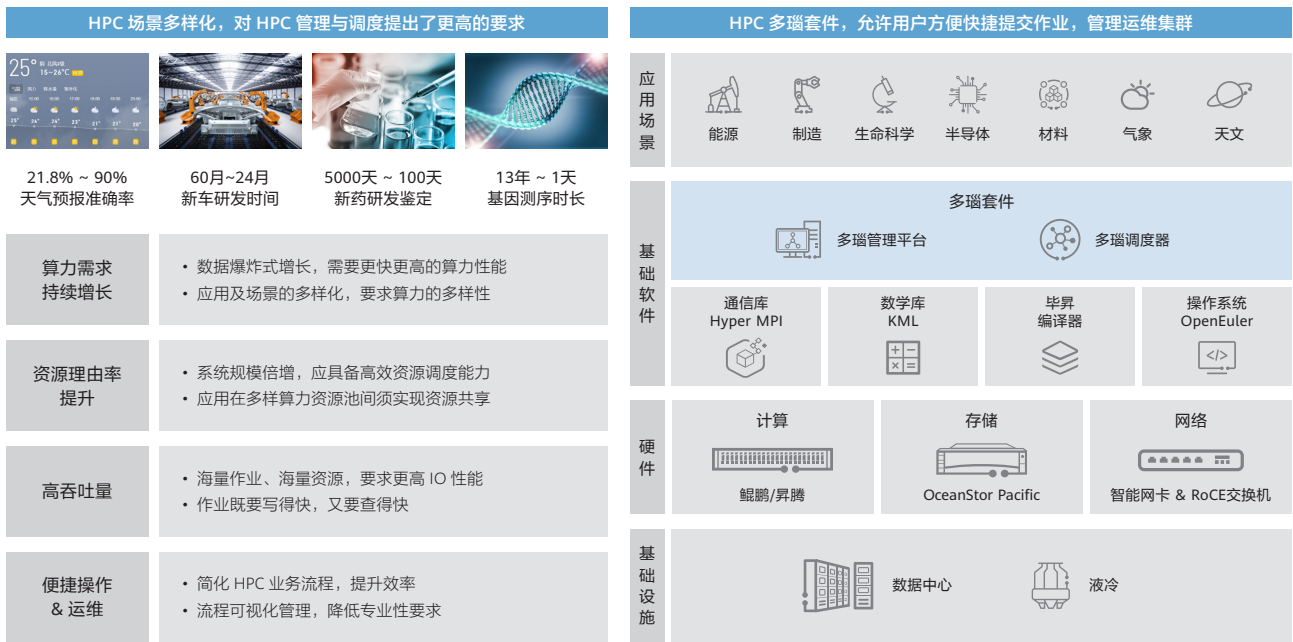
### KAE RSA 加解密

非对称加密算法 RSA，支持同步异步模式，支持 Key Sizes 1024/2048/3072/4096。在使用 Nginx 处理 HTTPS 请求时，可通过配置开启 SSL 模块，使能鲲鹏 RSA 加速引擎对 HTTPS 请求处理中 SSL/TLS 握手时的 RSA 非对称加解密运算进行加速。相比 x86 SSL 加速卡方案，鲲鹏 RSA 加速方案性能高 35%。

## HPC

HPC 聚焦资源调度效率低、应用性能优化难等关键挑战，通过全栈架构创新、软硬件自研、基础软件优化和行业应用性能调优等技术构建全栈高性能计算基础平台，帮助客户释放平台算力，缩短产品上市周期，提升企业产品竞争力。

鲲鹏 BoostKit HPC 应用使能套件，加速尖端科研创新和行业数字化



多瑙管理平台通过可视化界面为用户提供了便捷的 HPC 集群系统数据管理和软硬件资源管理功能，串联整个工作流程，帮助用户合理地进行作业调度和资源分配，提升集群系统计算能力利用率。

多瑙调度器提供大规模集群下的高资源利用率、高吞吐量的作业调度能力：

- » 超大规模调度：最大支持 3000 节点 /38 万核超大规模集群调度。
- » 高吞吐量作业：端到端吞吐量高，达到每小时 400 万+ 个作业。
- » 高资源分配率：高效灵活的调度框架，资源分配率达到 90%+。



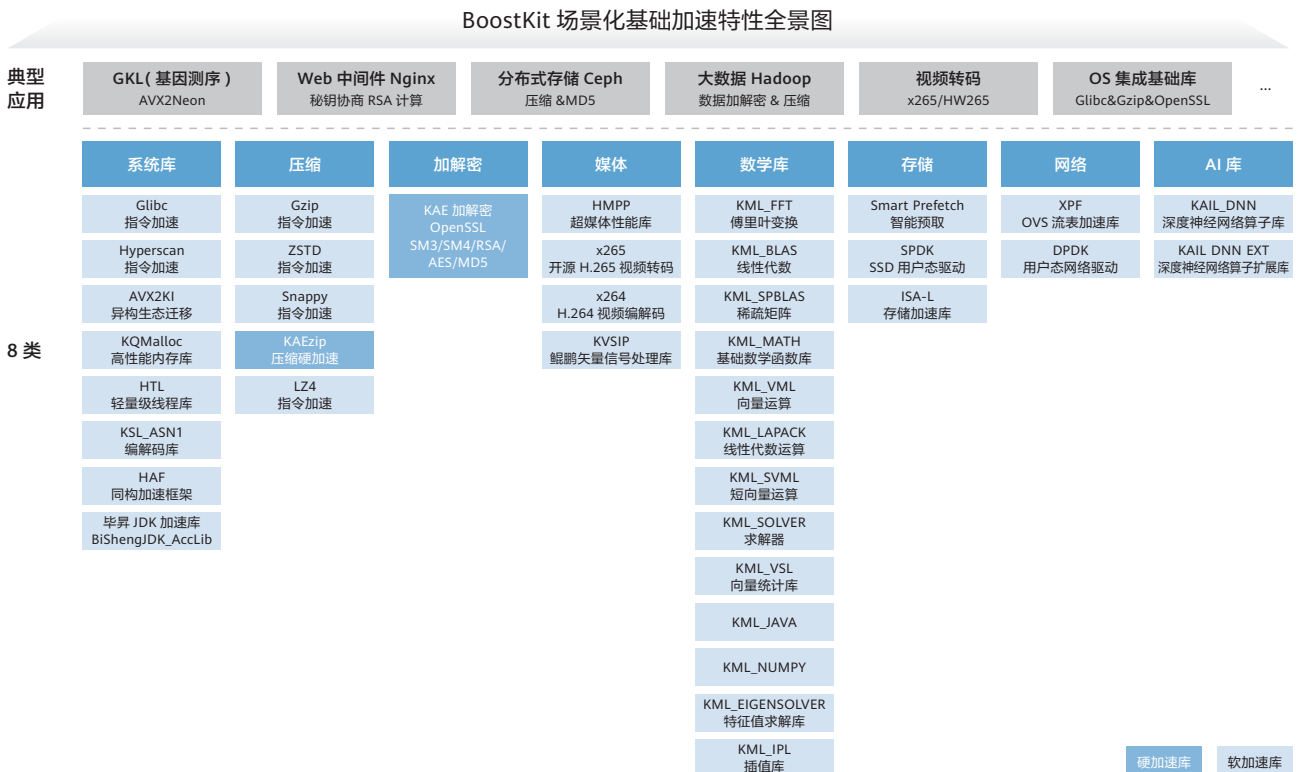
### 3 ) 鲲鹏 BoostKit 基础加速

鲲鹏 BoostKit 加速库提供基于 ARM 指令深度优化和基于鲲鹏 KAE (鲲鹏硬件加速引擎) 开发的加速库, 覆盖系统库、压缩、加解密、媒体、数学库、存储、网络和 AI 库等 8 类加速库, 为大数据加解密、分布式存储压缩、视频转码等应用场景提供高性能加速。

鲲鹏 BoostKit 基础加速软件包, 详细介绍和最新内容请参见[鲲鹏应用使能套件 BoostKit](#) → [基础加速软件包](#), 详情可扫描右方二维码。



图 3-19 鲲鹏 BoostKit 基础加速软件包



### 系统库

鲲鹏 BoostKit 系统库包括如下组件:

- » Glibc-patch: 主要对内存、字符串、锁等接口基于华为鲲鹏微架构特点进行了加速优化。微架构特点进行了加速优化。
- » Hyperscan: 一款高性能的正则表达式匹配库, 增加鲲鹏计算平台分支, 且完全兼容 armv8-a, 通过使用 NEON 指令、内联汇编、数据对齐、指令对齐、内存数据预取、静态分支预测、代码结构优化等方法, 实现在鲲鹏计算平台的性能提升。
- » AVX2KI: 一款接口集合库, 将传统平台的 Intrinsic 接口集合使用鲲鹏指令重新实现, 并封装为独立的接口模块 (C 语言头文件方式), 以减少大量迁移项目重复开发的工作量。
- » KQMAlloc: 鲲鹏高性能内存库, 是专为鲲鹏设计的内存分配器。此分配器分为单线程和多线程应用场景, 最大限度地减少内部缓存占用空间, 最大限度地减少内部缓存未命中, 可极大地提升应用性能。

- » HTL: 轻量级线程库 (Hyper Thread Library, HTL) 是构建在内核态线程之上的用户级线程库。用于解决内核级线程大量使用时, 应用性能下降、系统资源不足的问题, 尤其解决嵌套并行场景下性能极低的问题。在保障减少资源使用的情况下, 提升并发度, 改善性能。
- » KSL\_ASN1: ASN.1 (Abstract Syntax Notation One) 是定义抽象数据类型规格形式的一套标准, 是描述数据的表示、编码、传输、解码的灵活的一种记法。KSL\_ASN1 可替代开源 ASN.1 软件, 目前支持 PER (Packed Encoding Rules) 和 BER (Basic Encoding Rules) 格式, 性能相较开源软件 asn1c 有较大的提升。
- » HAF: 同构加速框架 (Homogeneous Acceleration Framework, HAF), 提供用户友好的编程方式和 API, 快速、有效、安全地使能用户业务程序指定加速片段卸载推送至卸载节点执行, 带来用户预期的卸载优化效果。
- » 毕昇 JDK 加速库: 毕昇 JDK 是基于 OpenJDK 开发的 Huawei JDK 开源版本, 基于鲲鹏处理器提供了一些加速能力, 如堆转储增强、JBooster 特性的加速。
  - 堆转储增强特性: 屏蔽转储堆文件中的敏感信息, 保护数据安全与隐私。
  - JBooster 特性: 提升应用启动速度、降低 CPU 占用、加快弹性伸缩的响应速度、降低云应用部署成本。
 其中, 堆转储增强特性支持毕昇 JDK 8 和 17 版本; JBooster 特性仅支持毕昇 JDK 17 版本。

## 压缩

鲲鹏 BoostKit 压缩库包括如下组件:

- » Gzip: Gzip (GUN zip) 通过数据预取、循环展开、CRC 指令替换等方法, 来提升其在鲲鹏计算平台上的压缩和解压缩速率, 尤其对文本类型文件的压缩及解压具有更明显的性能优势。
- » ZSTD: Zstandard, 即 ZSTD 压缩库。通过使用 NEON 指令、内联汇编、代码结构调整、内存预取、指令流水线排布优化等方法, 实现 ZSTD 在鲲鹏计算平台上压缩和解压性能的提升。
- » Snappy: 利用内联汇编、宽位指令、优化 CPU 流水线、内存预取等方法, 实现 Snappy 在鲲鹏计算平台上的压缩和解压速率提升。
- » KAEzip: 鲲鹏加速引擎的压缩模块, 使用鲲鹏硬加速模块实现 deflate 算法, 结合无损用户态驱动框架, 提供高性能 Gzip/zlib 格式压缩接口。
- » LZ4: 快速压缩库, 它能够以极高的速度对数据进行压缩和解压缩。该压缩算法非常快速和高效, 适用于各种类型的数据压缩。LZ4 库可以被广泛应用于数据存储、网络传输和实时数据处理等领域, 为用户提供了快速且可靠的数据压缩解决方案。其官网发布的 1.9.3 Release 优化版本, 利用 NEON 指令、内联汇编、代码结构调整、内存预取、指令流水线排布优化等方法, 实现 LZ4 在鲲鹏计算平台上压缩性能的提高。

## 加解密

KAE 加解密是鲲鹏加速引擎的加解密模块, 使用鲲鹏硬加速模块实现 RSA/SM3/SM4/DH/MD5/AES 算法, 结合无损用户态驱动框架, 提供高性能对称加解密、非对称加解密算法能力, 兼容 OpenSSL 1.1.1a 及其之后版本, 支持同步 & 异步机制。

## 媒体

鲲鹏 BoostKit 媒体库包括如下组件:

- » HMPP: 鲲鹏超媒体性能库 (Hyper Media Performance Primitives, HMPP), 包括向量缓冲区的分配与释放、向量初始化、向量数学运算与统计学运算、向量采样与向量变换、滤波函数、变换函数 (快速傅里叶变换), 支持 IEEE 754 浮点数运算标准。

- » x265: 针对 FFmpeg 视频转码场景, 对 X265 的转码底层算子使用鲲鹏向量指令进行加速优化, 提高整体性能。
- » x264: 采用 GPL 授权的视频编码免费软件, 主要功能实现 H.264/MPEG-4 AVC 的视频编码。
- » KVSIP: 鲲鹏矢量信号处理库, 提供了高性能计算接口, 包括向量基础运算、矩阵基础运算和快速傅里叶运算功能。

## 数学库

鲲鹏数学库 (Kunpeng Math Library, KML) 是华为提供的基于鲲鹏平台优化的高性能数学函数库, 主要完成标量、矢量、矩阵上的数学计算, 包括基本的四则运算、三角函数、对数函数、指数函数、线性代数计算等, 数学库所有接口由 C/C++、汇编语言实现, 部分接口兼容 Fortran 语言调用, 部分提供 Java 语言封装的接口。

## 存储

鲲鹏 BoostKit 存储库包括如下组件:

- » Smart Prefetch: 创新性地采用高速缓存配合高效的预取算法, 提升系统存储 IO 性能, 进而提升上述解决方案中存储 IO 密集型场景的整体性能。
- » SPDK: 高性能存储开发包 (Storage Performance Development Kit, SPDK) 通过使用网络技术、处理技术和存储技术来提升效率和性能。通过运行为硬件设计的软件, SPDK 已经证明很容易达到每秒钟数百万次 IO 读取, 通过使用许多处理器核心和许多 NVMe 驱动去存储, 而不需要额外卸载硬件。
- » ISA-L: ISA-L (Intelligent Storage Acceleration Library) 提供 RAID、纠删码、循环冗余检查、密码散列和压缩的高度优化的函数。

## 网络

鲲鹏 BoostKit 网络库包括如下组件:

- » XPF: XPF (Extensible Packet Framework) 在 OVS (Open vSwitch) 软件内部实现了一个智能卸载引擎模块, 该模块用于跟踪数据报文在 OVS 软件中所经历的所有流表和 CT 表, 将执行的 CT 行为和所有流表行为项进行综合编排成一条综合行为项并结合统一匹配项生成一条集成流表项。后续的数据报文在进入 OVS 后, 若匹配命中该集成流表, 则直接执行综合行为, 相比开源的处理流程, 查询次数将减少, 性能将大幅度提升。
- » DPDK: 数据平面开发工具包 (Data Plane Development Kit, DPDK), 为用户空间高效的数据包处理提供数据平面开发工具集, 包括库函数和驱动。

## AI 库

- » KAIL\_DNN: 深度神经网络算子库 (Deep Neural Network Library), 结合鲲鹏处理器微架构特性, 通过向量化、汇编、算法优化等手段, 提升 DNN 核心算子性能, 并通过插件化形式对接开源 oneDNN 库提供完整能力。
- » KAIL\_DNN\_EXT: 深度神经网络算子拓展库 (Deep Neural Network Extension Library), 旨在作为 KAIL\_DNN 的拓展库, 深度优化 softmax、random\_choice 等算子, 为 AI 特定场景封装为 Python 语言接口库直接提供给用户调用。

## 3.1.2 代码优化

### 3.1.2.1 代码优化介绍

鲲鹏原生开发在代码优化阶段提供了 DevKit 亲和分析工具，可实现鲲鹏亲和和代码优化建议，帮助开发者写出更适配鲲鹏平台的代码。

图 3-20 代码优化阶段



### 3.1.2.2 亲和分析工具

亲和分析工具是针对鲲鹏平台的开发者、用户和 ISV 开发者的应用、源码分析工具。亲和分析工具属于鲲鹏 DevKit 的扩展工具之一，支持在鲲鹏平台运行。支持以下功能：

- » 64 位运行模式检查：将原 32 位平台上的软件迁移到 64 位平台上，进行迁移检查并给出修改建议。
- » 字节对齐检查：在需要考虑字节对齐时，检查源码中结构体类型变量的字节对齐情况。
- » 缓存行对齐检查：对 C/C++ 源码中结构体变量进行 128 字节对齐检查，提升访存性能。
- » 内存一致性静态检查：在鲲鹏平台上，分析、修复用户软件中的内存一致性问题。
- » 向量化检查：在鲲鹏平台上检查可向量化片段并提供修改建议。
- » 矩阵化检查：在鲲鹏平台上检查可矩阵化片段并提供修改建议。
- » 构建亲和：分析 makefile、CMakeLists.txt 中可以替换成鲲鹏加速库的内容，并提供替换建议和功能修复。
- » 计算精度分析：精度分析工具对应用函数插桩后，分别在 x86 平台和鲲鹏平台运行，最终通过对比输出结果，分析平台计算精度差异。

快速集成	效率提示	性能倍增
支持 Jenkins、Gitlab 等业界主流流水线工具，快速集成 CI/CD 流水线	IDE 插件快速安装，提供开发态源码分析报告以及一键替换问题的能力	亲和鲲鹏硬件架构，全场景提升访存性能、计算精度

亲和分析工具详细介绍请参见[鲲鹏社区](#) → [鲲鹏开发条件 DevKit](#) → [应用迁移](#) → [有源码迁移](#) → [亲和分析](#)，详情可扫描右方二维码。



## 向量化检查

向量化检查功能用于对可向量化片段进行检查，并提供向量化修改建议。

### 步骤 1 创建向量化检查分析任务。

图 3-21 向量化检查（VS Code）



用户需要上传源码及编译生成的 BC 文件，BC 文件生成指导请参见联机帮助。

### 步骤 2 查看向量化检查分析结果。

图 3-22 向量化检查分析结果

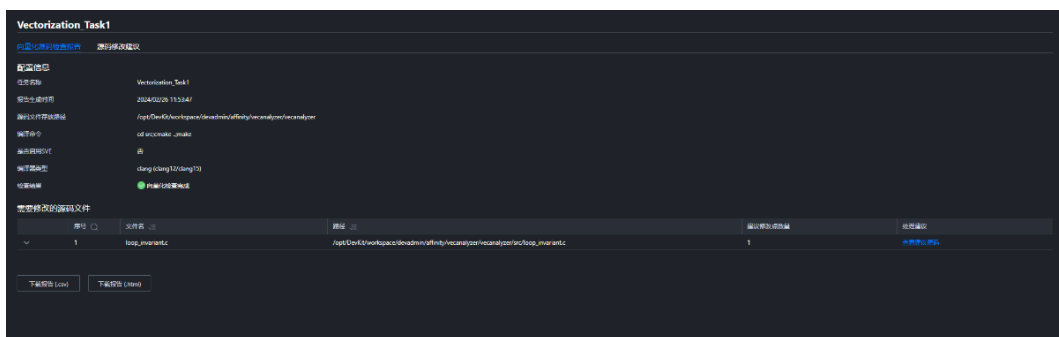


图 3-23 源码修改建议

```

C:\loop_invariant.c | X
C:\Users\kumpengfamily\kumpeng-dev-kit-24.0.1\resources\workspace\Vectorization_Task1\opt\DevKit\workspace\devadmin\affinity\vecanalyzer\vecanalyzer\src\C\loop_invariant.c
1  Suggestion: Please extract value outside the loop: loop_invariant.c:12:34
2  Example:
3  Before modification
4  for (i = 0; i < data->len; i++) {
5      vecC[i] = vecA[i] + vecB[i];
6  }
7
8  After modification
9  // Extract <data->len> outside the loop.
10 int len = data->len;
11 for (i = 0; i < len; i++) {
12     vecC[i] = vecA[i] + vecB[i];
13 }
14     array[i] += 1;
15 }
16 }
17 }
18 int main(void)
19 {
20     struct Data d;
21     int array[BUR_SIZE] = {0};
22     func(&d, array);
23     return 0;
24 }
25 }
26 }
27 }
28 }
    
```

---- 结束

## 3.1.3 编译

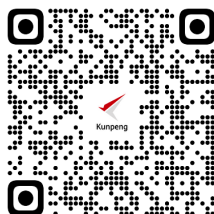
### 3.1.3.1 编译介绍

鲲鹏原生开发在编译构建阶段提供了毕昇编译器、毕昇 JDK、GCC for openEuler。通过上述鲲鹏亲和编译器，在不修改用户应用源码的情况下，提供鲲鹏平台的应用性能提升。

图 3-20 代码优化阶段



鲲鹏亲和编译器，详细介绍与最新内容请参见[鲲鹏社区](#) → [鲲鹏开发套件 DevKit](#) → [编译](#)，详情可扫描下方二维码。



### 3.1.3.2 毕昇编译器

毕昇编译器是华为编译器实验室针对通用处理器架构构建，增强和引入了多种编译优化技术，致力于打造高性能、高可信及易扩展的编译器工具链。

- » 支持 C11、C++14/17、Fortran 语言标准，OpenMP 4.5/5.0 API 标准。
- » 高性能编译算法，基于鲲鹏架构深度调优，内存优化、循环优化、自动向量化等多种编译增强。
- » 支持安全 / 高效编码工具集，包括静态检查工具、重构工具等。

### 3.1.3.3 毕昇 JDK

毕昇 JDK 基于 OpenJDK 开发，是一款高性能、可用于生产环境的 OpenJDK 发行版；毕昇 JDK 已应用于华为内部多个产品上，解决了业务实际运行中遇到的多个疑难问题；在 GC、鲲鹏后端代码生成、类信息共享等方面重点优化，在大数据等领域性能突出。毕昇 JDK 致力于为 JAVA 开发者提供一款稳定可靠、高性能、易调测的 JDK，也为用户在鲲鹏架构上提供一个更好的选择。

- » 支持 Java 版本：支持 Java 8、11、17、21 四个 LTS 版本。
- » 支持架构：Linux/AArch64、Linux/x86\_64。
- » 支持操作系统：openEuler 全系列操作系统、CentOS 7.6、Ubuntu 20.04、Ubuntu 22.04、麒麟 V10、UOS 20 等。

### 3.1.3.4 GCC for openEuler

GCC for openEuler 是基于开源 GCC 开发的编译器工具链（包含编译器、汇编器、链接器），在 openEuler 社区开源发布。

- » 支持 C11、C++14/17、Fortran 2018 语言标准，OpenMP 4.5/5.0 API 标准。
- » 针对鲲鹏平台进行质量加固，覆盖开源 + 商业全量测试，多产品稳定商用。



## 3.1.4 调试

### 3.1.4.1 调试介绍

鲲鹏编译调试工具支持在调试阶段进行应用调试。支持 C/C++/Fortran 代码调试能力、支持单机下 GPU 应用调试能力、支持 HPC 场景多节点应用并行调试能力。

图 3-25 调试阶段

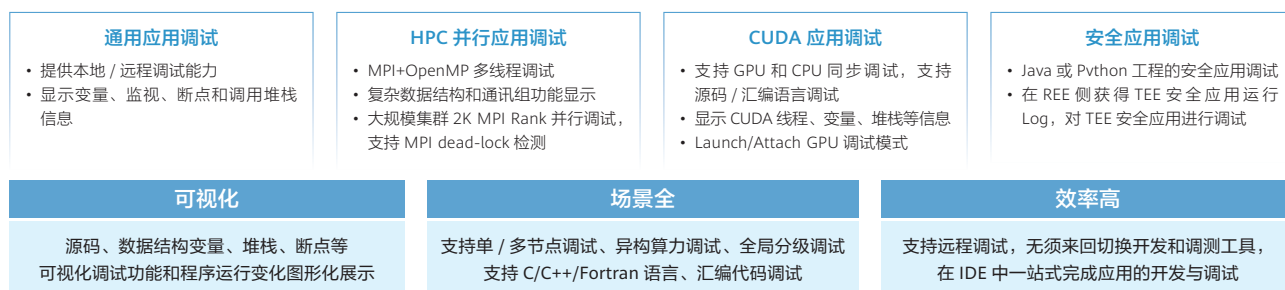


### 3.1.4.2 编译调试工具

编译调试工具能一键式部署鲲鹏编译器，支持单机下 GPU 应用调试能力，通过统一调试界面调试 GPU 应用，实现 CUDA-GDB 调试能力；支持 HPC 场景多节点应用并行调试能力以及鲲鹏平台远程编译调试能力。支持 C/C++/Fortran 代码调试能力。支持以下功能：

- » 一键式部署鲲鹏编译器（GCC for openEuler、毕昇编译器、毕昇 JDK）。
- » 可视化配置编译调试参数，一键式编译、运行、调试。
- » 远程单步调试 C/C++ 代码。
- » 编译调试过程中信息实时交互。
- » CUDA 支持调试源码、汇编代码。
- » HPC 支持调试源码、汇编代码，支持历史通信组详细信息展示和堆栈切换。

图 3-26 编译调试工具



编译调试工具，详细介绍与最新内容请参见[鲲鹏社区](#) → [鲲鹏开发套件 DevKit](#) → [调试](#)，详情可扫描右方二维码。

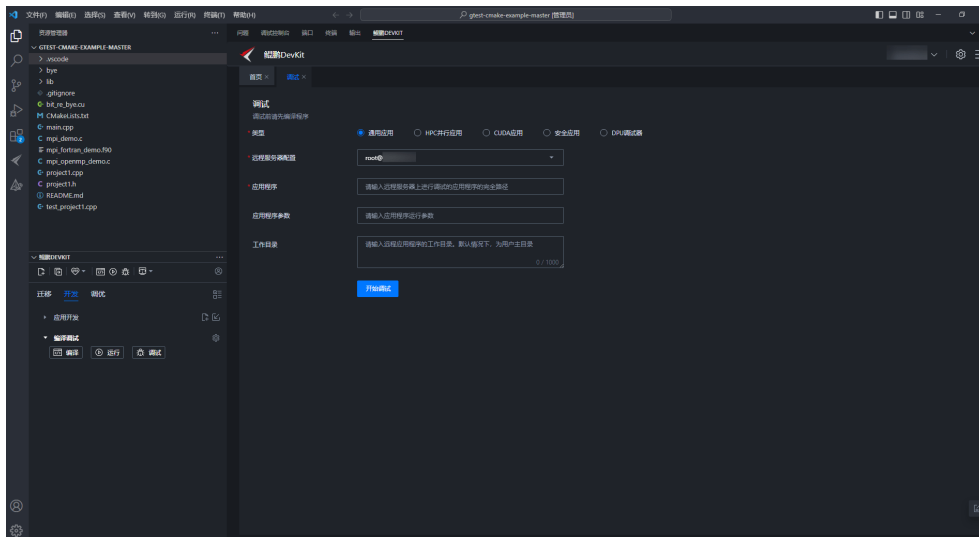


## 通用应用调试

通用应用调试提供鲲鹏、飞腾平台远程调试能力，提供图形化界面，大幅提升调试效率。

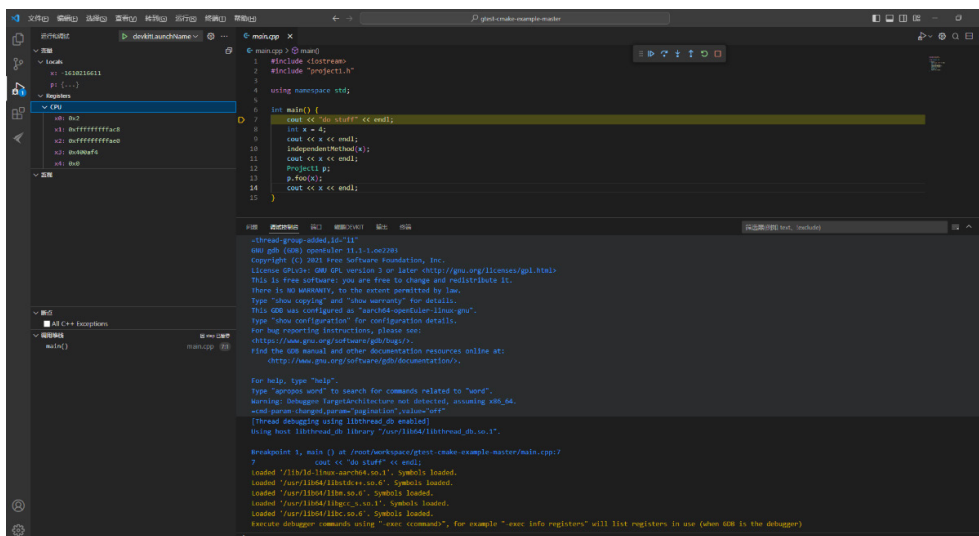
**步骤 1** 通用应用调试参数配置，如图 3-27 所示。

图 3-27 通用应用参数配置



**步骤 2** 进行通用应用调试，如图 3-28 所示。支持设置断点、单步调试、变量 / 寄存器查看等通用调试能力。

图 3-28 通用应用调试



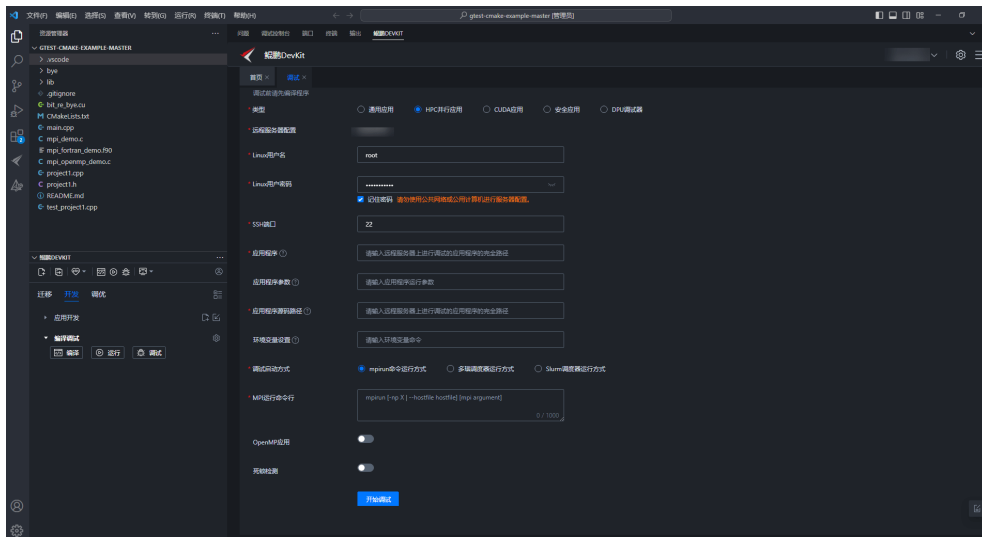
---- 结束

## HPC 并行应用调试

提供 HPC 场景多节点并行调试能力，仅支持以 Launch 模式调试 MPI 应用。

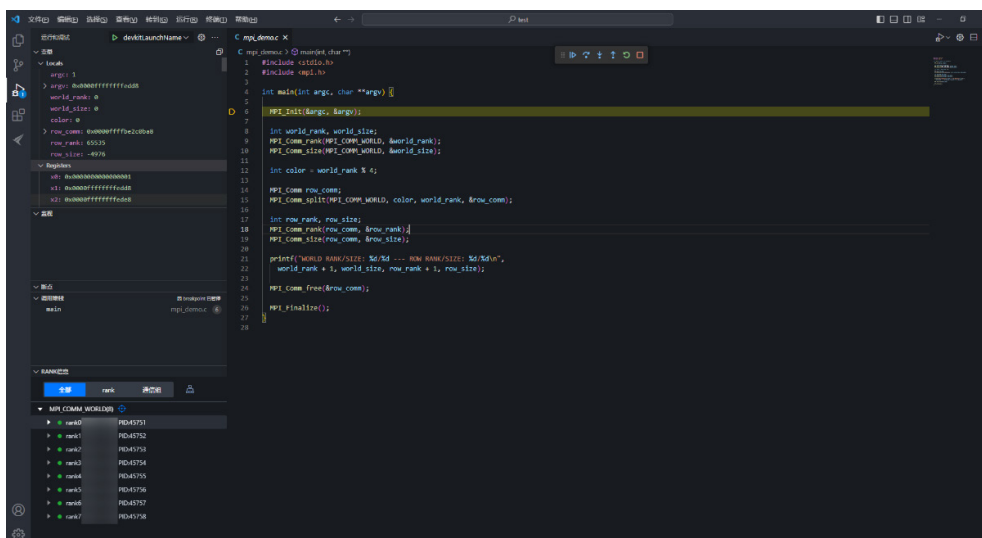
**步骤 1** HPC 并行应用调试参数配置，如图 3-29 所示。

图 3-29 HPC 并行应用参数配置



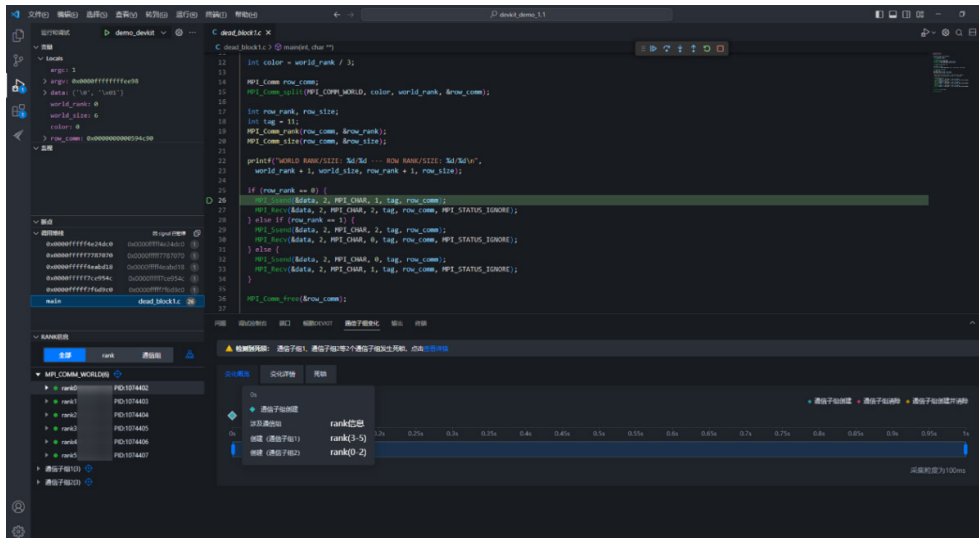
**步骤 2** 进行 HPC 并行应用调试，如图 3-30 所示。

图 3-30 HPC 并行应用调试



**步骤 3** 界面左下角呈现 RANK、通信组信息。单击  按钮，可查看通信子组变化概览、变化详情以及死锁信息，如图 3-31 所示。

图 3-31 查看详情



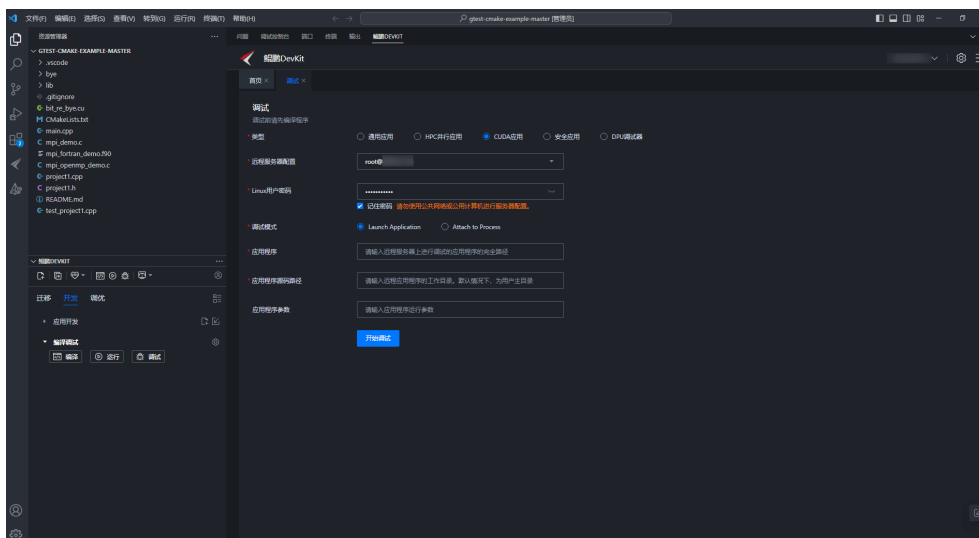
---- 结束

## CUDA 应用调试

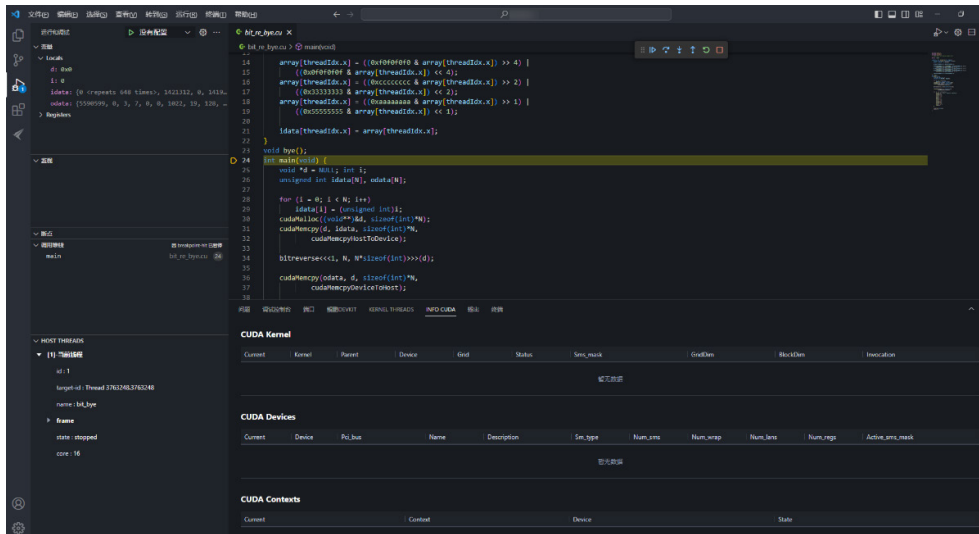
支持鲲鹏平台调试 CUDA 程序，通过统一的调试界面使用 CUDA-GDB 调试 GPU 应用。

**步骤 1** CUDA 应用调试参数配置，如图 3-32 所示。

图 3-32 CUDA 应用参数配置



步骤 2 进行 CUDA 应用调试，如图 3-33 所示。在源码下方可查看 KERNEL THREADS 信息以及 INFO CUDA 信息。  
图 3-33 CUDA 应用调试

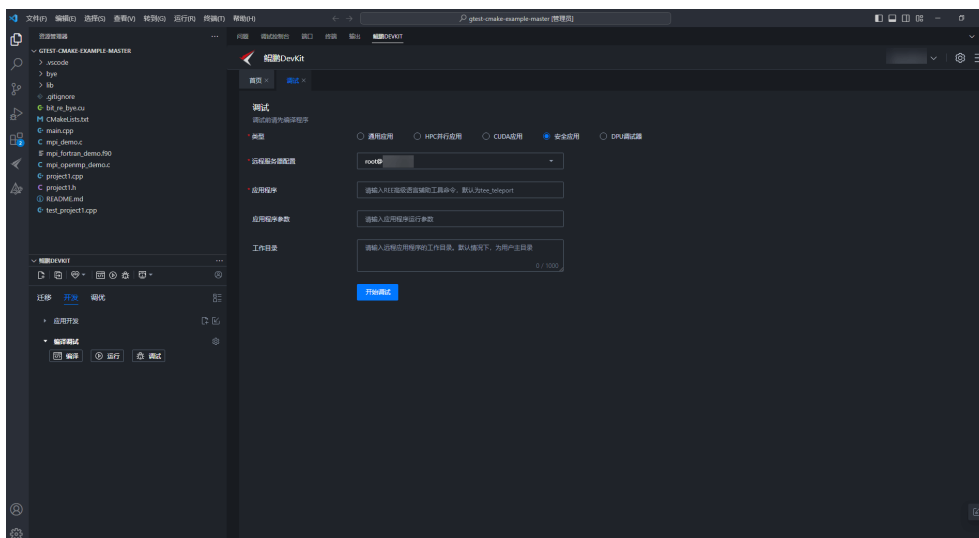


---- 结束

## 安全应用调试

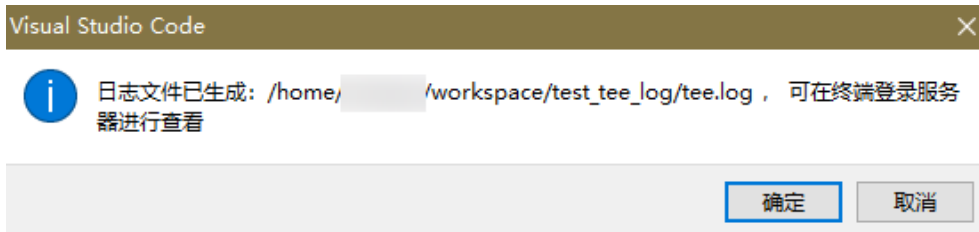
步骤 1 安全应用调试参数配置，如图 3-34 所示。

图 3-34 安全应用参数配置



步骤 2 通过查看 tee.log 查看调试结果，如图 3-35 所示。

图 3-35 日志文件



---- 结束

## 3.1.5 调优

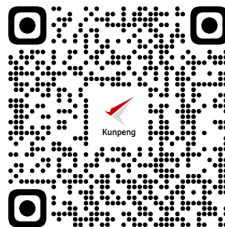
### 3.1.5.1 调优介绍

鲲鹏原生开发在调优阶段提供了系统性能分析工具、Java 性能分析工具、系统诊断工具。通过通用性能调优和场景化性能调优，可实现应用性能提升 5%-10%。

图 3-36 调优阶段



性能调优工具，详细介绍与最新内容请参见[鲲鹏社区](#) → [鲲鹏开发套件 DevKit](#) → [调优 & 诊断](#)，详情可扫描下方二维码。



### 3.1.5.2 系统性能分析工具

#### 系统性能分析工具简介

系统性能分析是针对基于鲲鹏的服务器的性能分析工具，能收集服务器的处理器硬件、操作系统、进程 / 线程、函数等各层次的性能数据，分析出系统性能指标，定位到系统瓶颈点及热点函数，给出优化建议。该工具可以辅助用户快速定位和处理软件性能问题。

调优助手是针对基于鲲鹏的服务器的调优工具，能系统化组织性能指标，引导用户分析性能瓶颈，实现快速调优。

图 3-37 系统性能分析工具

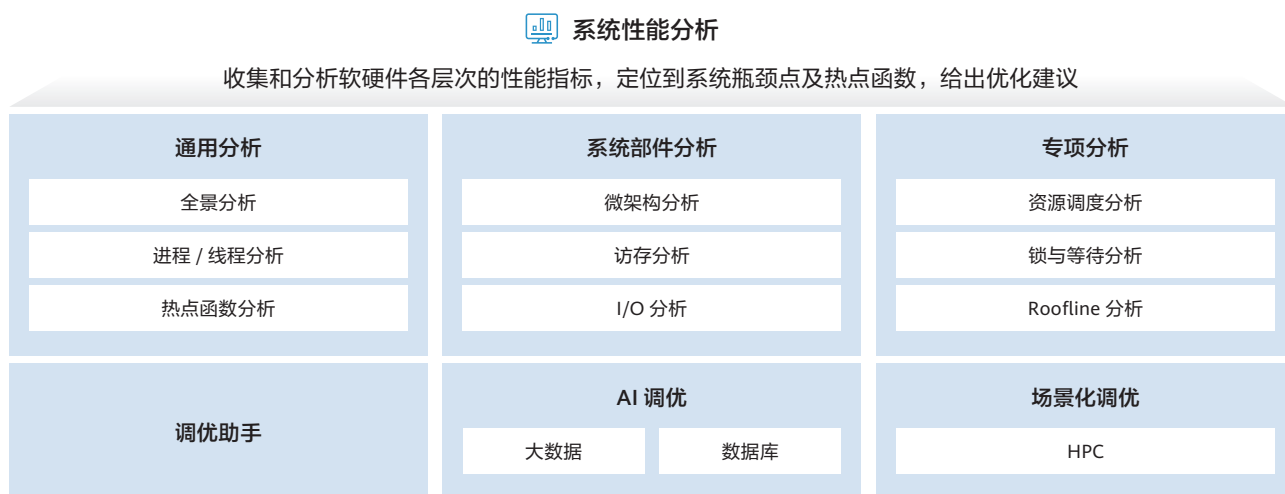


表 3-1 任务描述

任务分类	描述
调优助手	调优助手通过系统化组织和分析性能指标、热点函数、系统配置等信息，形成系统资源消耗链条，引导用户根据优化路径分析性能瓶颈，并针对每条优化路径给出优化建议和操作指导，以此实现快速调优。
对比分析	支持对同一种类型分析任务的结果，选择同一节点或者不同节点间进行比较，从而快速获得不同分析结果之间的差别，定位性能指标的变化，快速识别优化手段的效果。
HPC 集群检查	通过对用户指定的 MPI 集群进行硬件、软件配置检查，并给出集群中各节点软硬件配置的一致度报告。检查对象支持硬件领域的 CPU、GPU、互联、内存、网卡、磁盘，软件领域中的 OS、Kernel、环境变量、MPI、OpenMP、HPC 常用依赖库等维度。对于不符合鲲鹏平台最佳实践的配置，工具会给出优化建议。
HPC 应用分析	HPC 应用分析通过采集系统的 PMU 事件并配合采集面向 OpenMP 和 MPI 应用的关键指标，从而帮助用户精准获得 Parallel region 及 Barrier-to-Barrier 的串行及并行时间，校准的 L2 层微架构指标，指令分布及 L3 的利用率和内存带宽等信息。

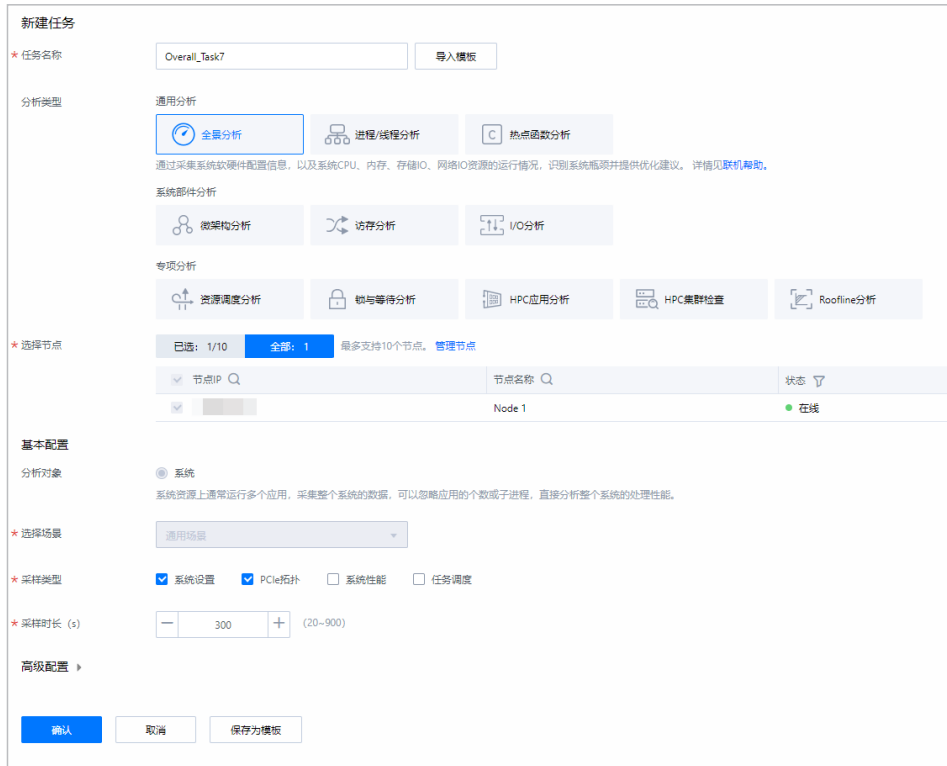
任务分类	描述
全景分析	通过采集系统软硬件配置信息，以及系统 CPU、内存、存储 IO、网络 IO 资源的运行情况，获得对应的使用率、饱和度、错误次数等指标，以此识别系统性能瓶颈。针对部分系统指标项，根据当前已有的基准值和优化经验提供优化建议。
微架构分析	基于 ARM PMU ( Performance Monitor Unit ) 事件，获得指令在 CPU 流水线上的运行情况，可以帮助用户快速定位当前应用在 CPU 上的性能瓶颈，用户可以有针对性地修改自己的程序，以充分利用当前的硬件资源。
访存分析	<p>基于 CPU 访问缓存和内存的事件，分析访存过程中可能的性能瓶颈，给出造成这些性能问题的可能原因及优化建议。</p> <p><b>访存统计分析</b>                      基于处理器访问缓存和内存的 PMU 事件，分析存储的访问次数、命中率、带宽等情况，具体包括：</p> <ul style="list-style-type: none"> <li>分析 L1C、L2C、L3C、TLB 的访问命中率和带宽。</li> <li>分析 HHA 访问速率。</li> <li>分析 DDR 的访问带宽和次数。</li> </ul> <p><b>Miss 事件分析</b>                      基于 ARM SPE ( Statistical Profiling Extension ) 能力实现。SPE 针对指令进行采样，同时记录一些触发事件的信息，包括精确的 PC 指针信息。利用 SPE 能力可以用于业务进行 LLC Miss, TLB Miss, Remote Access, Long Latency Load 等 Miss 类事件分析，并精确的关联到造成该事件的代码。基于这些信息，用户便可以有针对性地修改自己的程序，降低发生对应事件发生的几率，提高程序处理性能。</p> <p><b>NUMA 精细化分析</b>                      基于 ARM SPE ( Statistical Profiling Extension ) 能力实现。SPE 针对指令进行采样，同时记录一些触发事件的信息，包括精确的 PC 指针信息。利用 SPE 能力可以用于收集系统中所有进程的 NUMA 性能，找到 Top N (e.g. N = 10) NUMA 性能最差的进程及这些进程中的内存热区，各 NUMA 节点间内存访问统计矩阵，识别节点间内存访问不平衡状态，并得到相关优化建议。</p>
I/O 分析	分析存储 IO 性能。以存储块设备为分析对象，分析得出块设备的 I/O 操作次数、I/O 数据大小、I/O 队列深度、I/O 操作时延等性能数据，并关联到造成这些 I/O 性能数据的具体 I/O 操作事件、进程 / 线程、调用栈、应用层 I/O APIs 等信息。根据 I/O 性能数据分析给出进一步优化建议。
进程 / 线程性能分析	采集进程 / 线程对 CPU、内存、存储 IO 等资源的消耗情况，获得对应的使用率、饱和度、错误次数等指标，以此识别进程 / 线程性能瓶颈。针对部分指标项，根据当前已有的基准值和优化经验提供优化建议。针对单个进程，还支持分析它的系统调用情况。
资源调度分析	采集进程 / 线程的运行情况，获得对应的冷火焰图、链路切换次数和全局占比等指标，以此识别性能瓶颈。支持分析单个进程的系统调用情况。
热点函数分析	分析 C/C++ 程序代码，找出性能瓶颈点，获得对应的热点函数，支持通过火焰图展示函数的调用关系，给出优化路径。
锁与等待分析	分析 glibc 和开源软件 ( 如 MySQL、Open MP ) 的锁与等待函数 ( 包括 sleep、usleep、mutex、cond、spinlock、rwlock、semaphore 等 )，关联到其归属的进程和调用点，并根据当前已有的优化经验给出优化建议。
Roofline 分析	帮助用户在给定硬件平台上，分析出应用程序的瓶颈点位置，从而有针对性的进行优化。
AI 调优	使用自研高性能 AI 调优方案，通过用户自主选择测试用例，对数据库和大数据场景进行自动优化，自动调优后给予最优参数配置，提供复杂场景下参数配置的优化建议。

## 通用分析

通用分析包含全景分析、进程 / 线程分析、热点函数分析。

**步骤 1** 创建通用分析任务。如图 3-38 所示。

图 3-38 通用分析任务



**步骤 2** 查看任务分析结果，如图 3-39、图 3-40 和图 3-41 所示。

图 3-39 全景分析结果

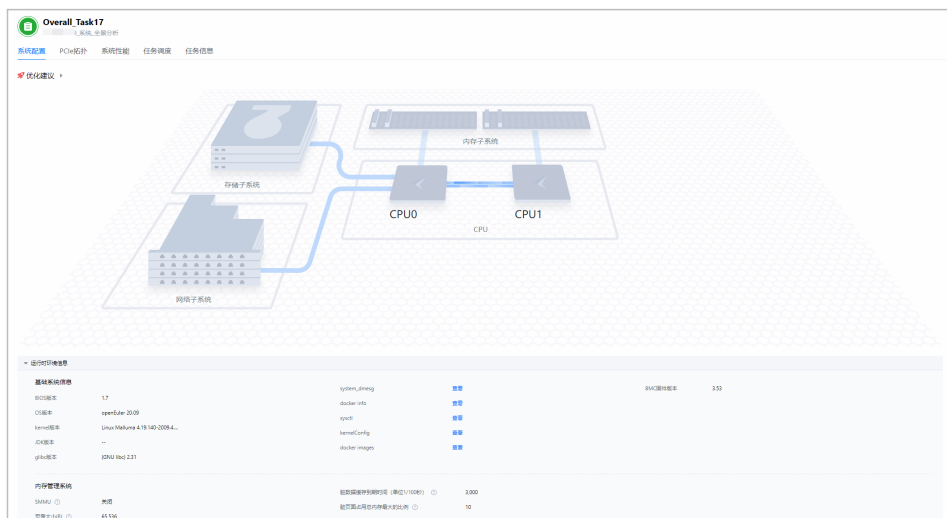


图 3-40 进程 / 线程分析结果

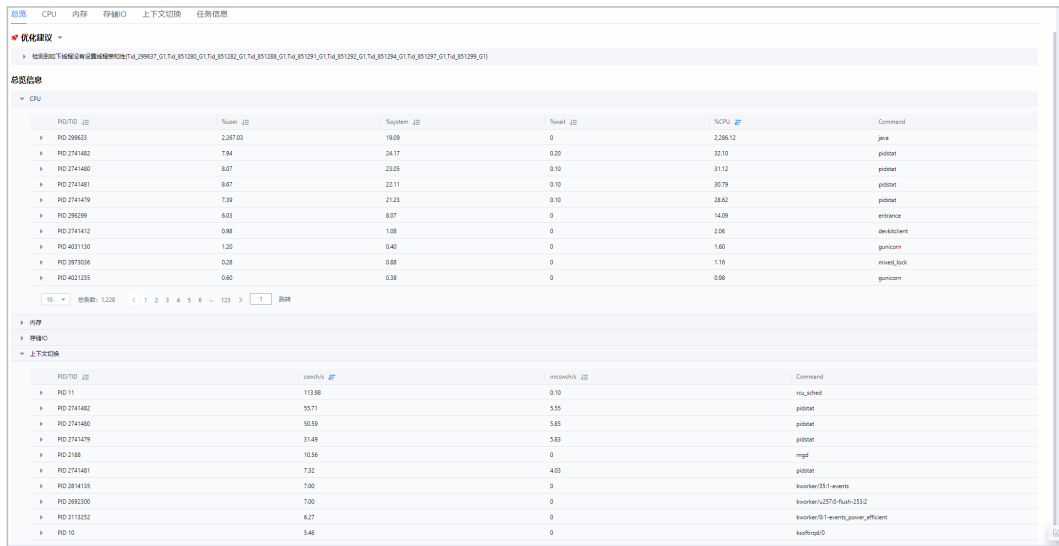
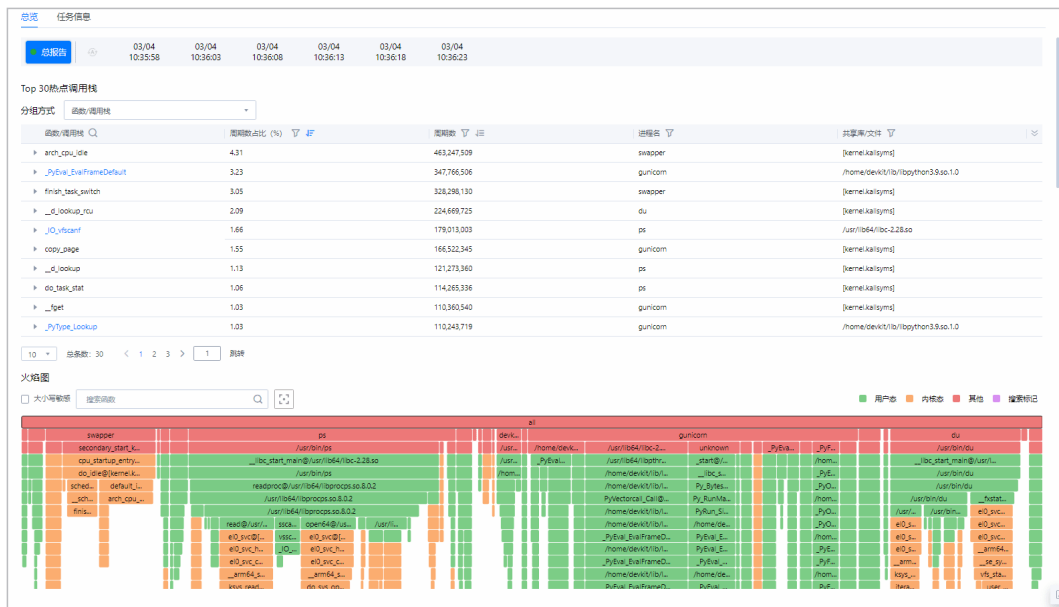


图 3-41 热点函数分析结果



---- 结束

## 系统部件分析

系统部件分析包括微架构分析、访存分析、I/O 分析。

**步骤 1** 创建系统部件分析，如图 3-42 所示。

图 3-42 系统部件分析

**步骤 2** 查看分析结果，如图 3-43、图 3-44 和图 3-45 所示。

图 3-43 微架构分析结果

事件名称	事件比例
Pipeline Slots	100.00 %
Backend Bound	19.32 %
Bad Speculation	15.10 %
Frontend Bound	55.10 %
Retiring	10.48 %



## 专项分析

专项分析包括资源调度分析、锁与等待分析和 Roofline 分析。

**步骤 1** 创建专项分析任务，如图 3-46 所示。

图 3-46 专项分析

### 新建任务

\* 任务名称:  导入模板

分析类型

通用分析

全景分析  进程/线程分析  热点函数分析

系统部件分析

微架构分析  访存分析  I/O分析

专项分析

资源调度分析  锁与等待分析  HPC应用分析  HPC集群检查  Roofline分析

基于CPU调度事件分析系统资源调度情况。详情见联机帮助。

\* 选择节点

已选: 1/10 全部: 1 最多支持10个节点。管理节点

节点IP	节点名称	状态
<input type="checkbox"/>	Node 1	● 在线

基本配置

分析对象

系统  应用

系统资源上通常运行多个应用。采集整个系统的数据，可以忽略应用的个数或子进程，直接分析整个系统的处理性能。

\* 采样时长 (s)

(1~1,800)

随着采样时长增加，采集处理可能会因超过采集空间大小而终止，最大支持10GB数据采集。

高级配置 >

确认 取消 保存为模板

**步骤 2** 查看分析结果，如图 3-47、图 3-48 和图 3-49 所示。

图 3-47 资源调度分析结果

进程/线程ID	PID	QID	PPID	PPID	运行时间 (ms)	阻塞时间 (ms)	等待消息的时间 (ms)	最大消息的时间 (ms)	最大消息的时间 (ms)	操作
rsched	219500		219500		2147414	72340	0.000	1229	1.617507	查看
rsched/1238	421410		421410		128307	12379	0.000	4000	8.492540	查看
rsched/1239	420620		421030		107296	15140	0.000	1360	9.217640	查看
rsched/1237	420160		420160		106181	14387	0.004	1340	1.439937	查看
rsched/1237	423910		423910		140339	13779	0.004	2410	2.432382	查看
rsched/1236	421100		421100		85311	8338	0.000	6020	0.462340	查看
rsched/1230	421724		421724		46523	7218	0.000	6200	0.205551	查看
rsched/1280	424201		424201		28270	8303	0.000	6619	0.207301	查看
rsched/1280	422187		[unknown]		26264	6166	0.000	6020	0.194099	查看
rsched/1284	422187		422187		48377	4248	0.000	6275	1.041482	查看

NUMA节点ID	PID	QID	PPID	PPID	阻塞时间 (ms)	操作
rsched	219500		219500		32	查看
rsched	219500		219500		24	查看
rsched	219504		219500		24	查看
rsched	219507		219500		24	查看
rsched	219510		219500		20	查看
rsched	219509		219500		18	查看
rsched	219501		219500		18	查看
rsched	219504		219500		18	查看
rsched	219501		219500		18	查看

图 3-48 锁与等待分析结果

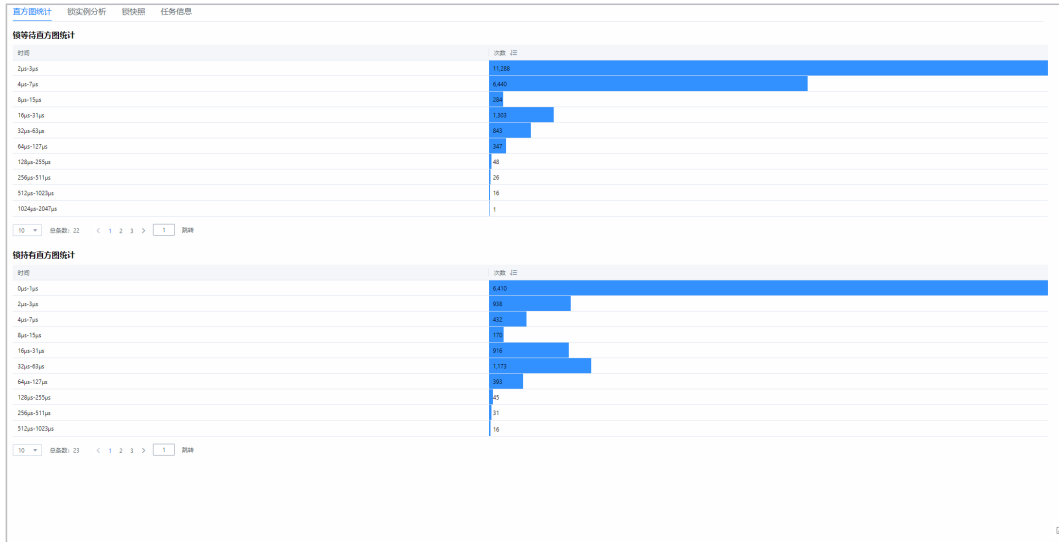
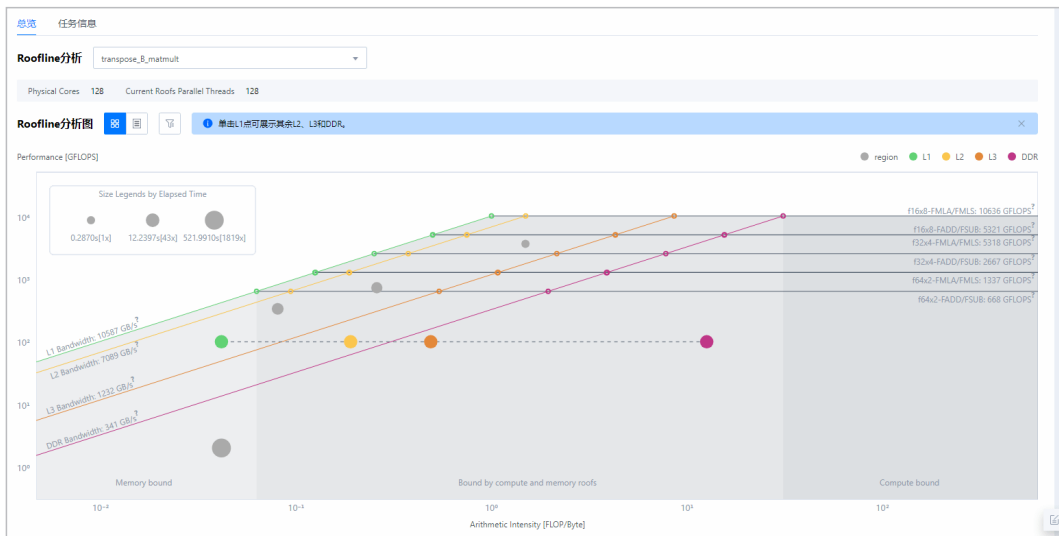


图 3-49 Roofline 分析结果



---- 结束

## 调优助手

步骤 1 创建调优助手分析任务，如图 3-50 所示。

图 3-50 调优助手分析任务

The screenshot shows the 'TuningAssistant\_Task1' configuration window. It includes fields for task name, node selection (Node 1), analysis target (Application), mode (Launch application), application path, application parameters, application user, symbol file path, sampling time (15s), and collection file size (100 MIB). A '立即分析' (Analyze Now) button is at the bottom.

步骤 2 查看分析结果，如图 3-51、图 3-52、图 3-53 和图 3-54 所示。

图 3-51 系统配置分析结果

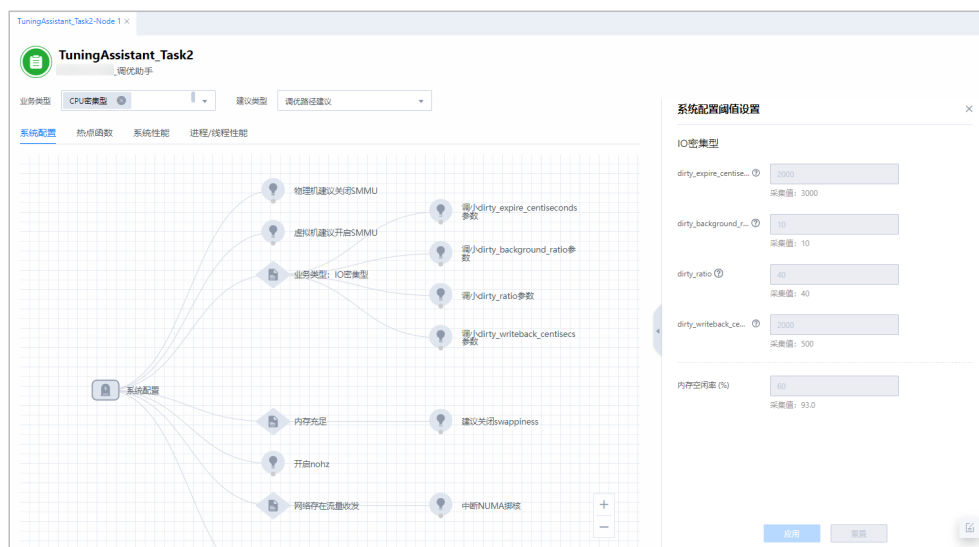


图 3-52 热点函数分析结果

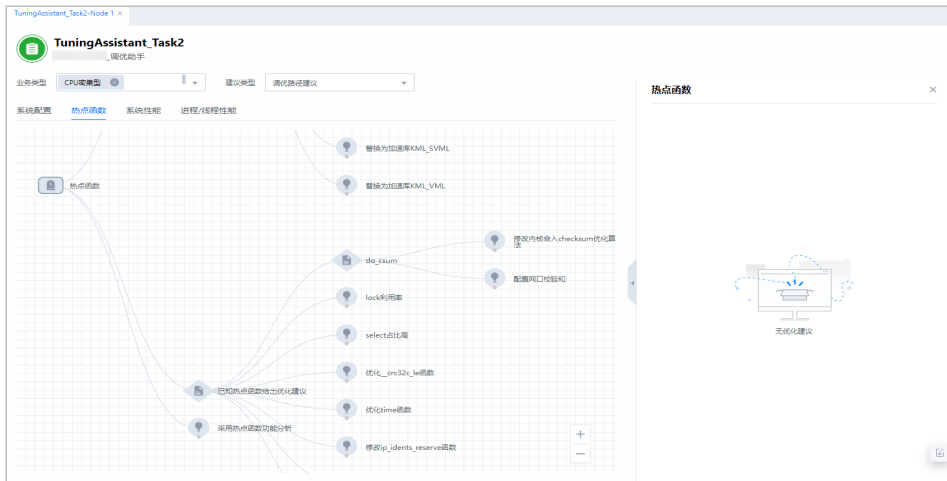


图 3-53 系统性能分析结果

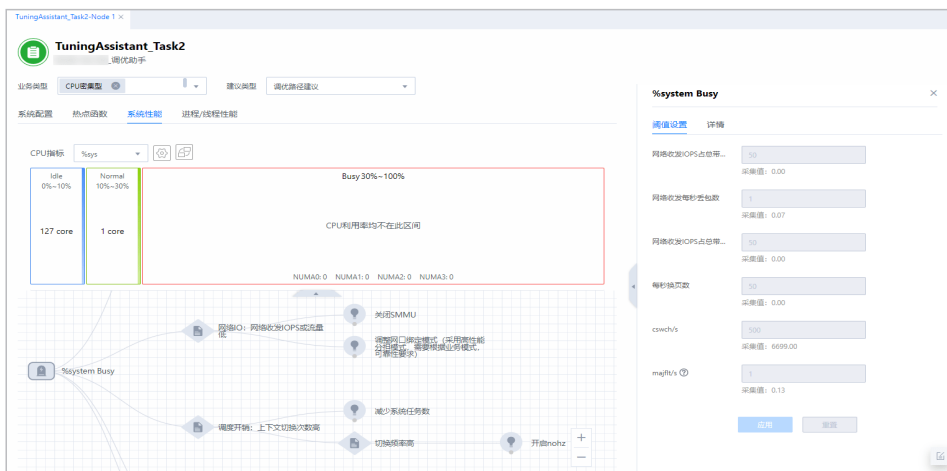
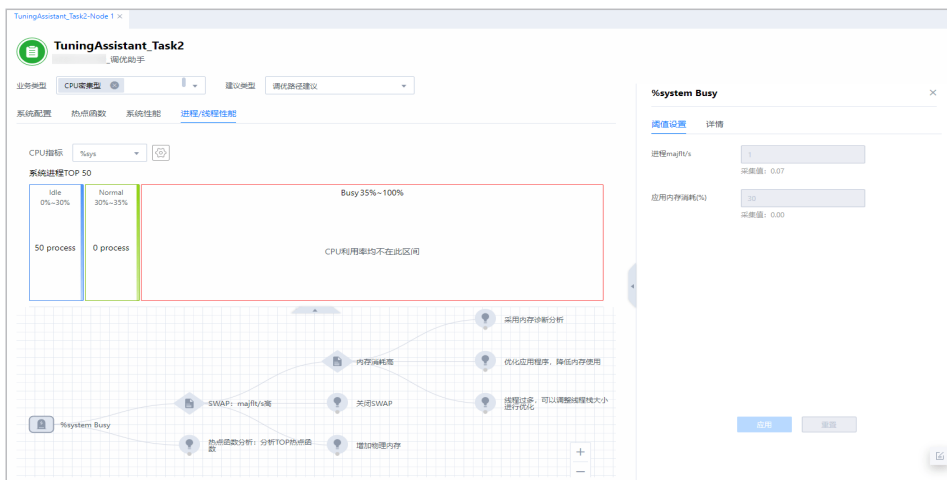


图 3-54 进程 / 线程性能分析结果



---- 结束

## AI 调优

步骤 1 创建 AI 调优任务，如图 3-55 所示。

图 3-55 AI 调优任务

新建AI调优分析任务 ×

### 新建AI调优分析任务

\* 任务名称: MySQL\_Task1

\* 应用分类: 数据库

\* 应用名称: MySQL

\* 应用版本: 5.7.22

\* root用户密码: 请输入DevKit节点的root密码  
 记住密码

**配置目标节点**

\* 压测工具所在节点: 请选择压测工具所在节点

\* MySQL所在节点: 请选择目标应用所在节点

**配置环境变量**

libmysqlclient.so路径: 例: /application/mysql/lib

**配置目标应用**

\* mysql执行文件路径: 例: /application/mysql/bin  
 mysqlid与mysql执行文件路径一致

\* mysqld执行文件路径: 例: /application/mysql/bin

\* 应用账号: 请输入应用执行用户

\* 配置文件路径: /etc/my.cnf

\* 应用配置参数:

应用配置参数	应用配置参数范围	初始值	操作
<input checked="" type="checkbox"/> innodb_adaptive_flushing_lwm	0~70	10	<a href="#">修改</a>
<input checked="" type="checkbox"/> innodb_adaptive_max_sleep_delay	0~1000000	150000	<a href="#">修改</a>
<input checked="" type="checkbox"/> innodb_max_dirty_pages_pct_lwm	0~99	10	<a href="#">修改</a>
<input checked="" type="checkbox"/> thread_cache_size	2000~16384	2000	<a href="#">修改</a>
<input checked="" type="checkbox"/> innodb_io_capacity_max	10000~160000	30000	<a href="#">修改</a>
<input checked="" type="checkbox"/> innodb_io_capacity	28000~160000	30000	<a href="#">修改</a>
<input checked="" type="checkbox"/> innodb_flushing_avg_loops	1~60	30	<a href="#">修改</a>
<input checked="" type="checkbox"/> innodb_sync_spin_loops	0~100	5	<a href="#">修改</a>
<input checked="" type="checkbox"/> innodb_spin_wait_delay	0~50	6	<a href="#">修改</a>
<input checked="" type="checkbox"/> query_alloc_block_size	4096~65536	4096	<a href="#">修改</a>

10 总数: 33 < 1 2 3 4 > 1 刷新

● 请确认MySQL处于开启状态，且正确配置MySQL密码。

**压测配置**

\* 压测工具: sysbench

\* 压测工具版本: 0.5

\* 测试用例: oltp\_mix

\* oltp.lua路径: 例: /opt/sysbench-0.5/tests/db

\* 调优性能指标: tps

\* 数据库: 请输入数据库名称

\* 压测工具路径: 例: /opt/sysbench-0.5/sysbench

\* 线程数:

\* 调优迭代次数: 150  
 ● 随着调优迭代次数增加，AI调优处理时长会增加。

步骤 2 调优结束后，可通过单击“下载优化后参数集”按钮下载已调优的参数配置。

图 3-56 下载优化后参数集



---- 结束



## 场景化调优

步骤 1 创建场景化调优任务，如图 3-57 所示。

图 3-57 场景化调优任务

**新建任务**

\* 任务名称: Open\_MP\_MPI\_Task7 [导入模板]

分析类型: 通用分析 (全景分析, 进程/线程分析, 热点函数分析), 系统组件分析 (架构分析, 访存分析, I/O分析), 专项分析 (资源调査分析, 锁与等待分析, **HPC应用分析**, HPC集群检查, Roofline分析)

通过采集系统的PMU事件, 获取面向OpenMP和MPI应用的关键指标, 以及内存常量、指令分布、微架构指标等信息。详见[联机帮助](#)。

\* 选择节点: 已选: 1/10 全部: 1 最多支持10个节点。管理节点

节点IP: [ ] 节点名称: [ ] 状态: [ ]

Node 1 [ ] 在线

**基本配置**

分析对象:  应用  
应用运行在系统上, 采集应用使用的系统资源数据, 分析应用的性能。

模式:  Launch application  
启动采集任务时同时启动应用, 采样时长由应用的执行时间决定, 适用于应用运行时间较短的场景。

应用运行用户:  应用默认运行在工具内置devkworker1用户下, 且仅支持应用在/opt或/home路径下执行; 如果应用运行对用户有依赖, 需打开选项配置对应的用户密码后在对应用户下执行。

\* 应用路径: [ ]  
请输入应用所在的绝对路径。  
注意: 请确保应用无安全风险。  
当前仅支持应用在/opt或/home路径下执行, 如有其他路径执行需求, 请开启应用运行用户并在系统设置的应用程序路径配置项增加需求路径。

应用参数: [ ]  
请输入应用参数

\* 采集模式:  OpenMP模式  MPI / MPI + OpenMP混合模式

\* 分析模式:  统计分析  精细化分析

**采样配置**

\* 采样模式:  Summary模式  Detail模式  
Summary模式采集基础和指标, 采集分析开销较小; Detail模式采集热点函数及详细指标, 采集分析开销较大。

\* 采样时长 (s): [ 300 ] (1-600)  
随着采样时长增加, 采集处理可能会因超过设定的采集数据大小而终止。

\* 延迟采样时长 (s): [ 0 ] (0-900)  
用于指定时间后执行分析, 可以忽略程序的启动流程分析或用于采集程序热身, 消除环境检测等带来的采集延迟。

\* 任务时间:  立即执行  预约定时启动

[ 确认 ] [ 取消 ] [ 保存为模板 ]

图 3-58 HPC 应用分析总览

系统: 8871510 自身信息

\* 优化建议

1. 检查CPU使用率, CPU使用率过高, 可能会导致性能下降, 请检查CPU使用率, 避免CPU使用率过高。

2. 检查内存使用率, 内存使用率过高, 可能会导致性能下降, 请检查内存使用率, 避免内存使用率过高。

3. 检查I/O使用率, I/O使用率过高, 可能会导致性能下降, 请检查I/O使用率, 避免I/O使用率过高。

应用ID: 304-3778-1666-000-1000-300-7916

CPU使用率: 10.8% / 100.0% (not set) / 100.0%

CPU: 0.421% [ ]

Memory: 16.6% / 100.0% / 100.0%

**内存使用**

内存使用	内存分布
平均内存使用率: 2.4954 GB/s	1.7 G/s Peak 100%
应用内存: 0.8500 GB/s	13.26 %
系统内存: 0.1056 GB/s	4.26 %
Socket内存使用率: 0.4242 GB/s	17.11 %
Cache内存使用率: 0.5262 GB/s	21.10 %
	Memory: 45.51 %
	Integer: 33.21 %
	Floating Point: 0.00 %
	Advanced SIMD: 0.00 %
	Not Retired: 17.22 %

**HPC Top-Down**

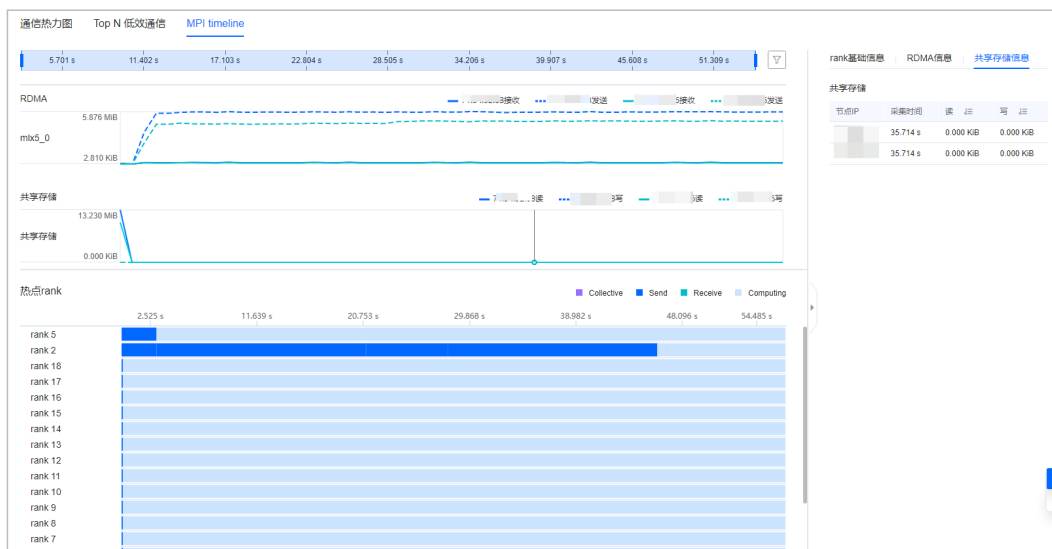
事件名称	占比
Rating	77.91 %
Retired Bound	5.88 %
Memory Bound	5.69 %
L1 Bound	0.90 %
L2 Bound	0.90 %
L3 or DSM Bound	0.90 %
Non Bound	0.90 %
Cache Bound	5.69 %
Forward Bound	2.62 %
Not Speculation	1.00 %

**MPI应用指标**

应用ID: [ ]

步骤 3 查看 MPI timeline，如图 3-59 所示。

图 3-59 MPI timeline



---- 结束

### 3.1.5.3 Java 性能分析工具

#### Java 性能分析工具简介

Java 性能分析工具是针对基于鲲鹏的服务器上运行的 Java 程序的性能分析和优化工具，能图形化显示 Java 程序的堆、线程、锁、垃圾回收等信息，收集热点函数、定位程序瓶颈点，帮助用户采取针对性优化。

图 3-60 Java 性能分析工具



表 3-1 任务描述

任务分类	描述
在线分析	<p>在线分析包含对于目标 JVM 和 Java 程序的双重分析。包括 Java 虚拟机的内部状态如 Heap, GC 活动, 线程状态及上层 Java 程序的性能分析, 如调用链分析, 热点函数, 锁分析, 程序线程状态及对象生成分布等。通过 Agent 的方式在线获取 JVM 运行数据, 进行精确分析。</p> <p>主要分析结果包含:</p> <ol style="list-style-type: none"> <li><b>1. 概览</b> <ul style="list-style-type: none"> <li>在线显示 Java 虚拟机系统状态。</li> <li>在线显示 JVM 的 Heap 大小、GC 活动、Thread 数量、Class 加载数量和 CPU 使用率。</li> </ul> </li> <li><b>2. 线程信息</b> <p>获取当前 JVM 中实时的活动线程状态和当前线程转储, 图形化显示线程锁定状态, 分析线程死锁情况。</p> </li> <li><b>3. 内存信息</b> <ul style="list-style-type: none"> <li>通过抓取堆快照, 分析应用在某时刻堆的直方图分布和支配调用关系, 追溯堆内存中各 Java 存活对象到 GC root 的引用关系链, 帮助定位潜在的内存问题; 对比分析不同时刻的堆快照, 给出堆使用与分配变化, 辅助用户发现堆内存在分配和使用过程中的异常情况。</li> <li>获取 Java 堆中各个对象创建的数量及大小, 显示相关内存使用情况并实时刷新。</li> </ul> </li> <li><b>4. 热点信息</b> <ul style="list-style-type: none"> <li>通过工具分析的热点方法, 热点方法以倒火焰图形式呈现, 不同层 (如 Java 调用层、JNI 层、Native 层、内核层等) 的热点方法的以不同颜色区分。</li> <li>通过工具展开查看 Java 方法对应的字节码 (非必须)、经 JVM JIT 编译器生成的机器指令, 并查看这些指令的热点分布; 对于不能查看的字节码的情形需要给出原因。</li> </ul> <p>对给定的入口方法进行调用链采集, 收集采集期间的方法调用关系, 方法耗时等数据; 以树形式展现。</p> </li> <li><b>5. GC 信息</b> <p>用户可以在线统计分析目标 JVM 进程中发生的 GC 事件, 分析 GC 原因、GC 阶段分布、GC 内存回收效果、GC 暂停等因素, 以定位潜在的 GC 相关的内存问题、性能瓶颈问题。</p> </li> <li><b>6. IO 信息</b> <p>在线分析应用中的文件 IO、Socket IO 时延、消耗带宽等数据, 找出热点 IO 操作。</p> </li> <li><b>7. 数据库信息</b> <ul style="list-style-type: none"> <li>监测和分析数据库连接池。监测数据库连接池连接的情况、帮助用户定位潜在的连接泄露, 对不合适的连接池配置给出优化建议。</li> <li>分析 JDBC 热点 SQL 操作。记录应用中的 SQL 调用时间、耗时和堆栈跟踪, 帮助用户定位耗时最长的热点 SQL 操作。</li> <li>分析 NoSQL 热点操作。记录应用中访问 / 操作 NoSQL 数据库调用时间、耗时和堆栈跟踪, 帮助用户定位耗时最长的热点 NoSQL 操作。</li> </ul> </li> <li><b>8. HTTP 信息</b> <p>记录应用中的 HTTP 请求时间和耗时, 找出热点 HTTP 请求。</p> </li> <li><b>9. 快照信息</b> <p>支持在堆、IO、Workload 在线分析过程中生成快照, 对快照进行比对, 辅助用户发现资源、业务相关指标的变化趋势, 定位潜在的资源泄露问题或性能指标恶化问题。</p> </li> </ol>

任务分类	描述
采样分析	<p>通过采样的方式，收集 JVM 的内部活动 / 性能事件，通过录制及回放的方式来进行离线分析。这种方式对系统的额外开销很小，对业务影响不大，适用于大型的 Java 程序。</p> <p>主要分析结果包括：</p> <ol style="list-style-type: none"> <li><b>1. 概览</b> <ul style="list-style-type: none"> <li>显示 Java 虚拟机系统状态。</li> <li>通过采样及回放的方式显示 JVM 的 Heap 使用情况、GC 活动、IO 消耗和 CPU 使用率。</li> </ul> </li> <li><b>2. 线程转储及锁分析结果</b> <ul style="list-style-type: none"> <li>分析程序线程状态及锁。获取采样时间内的线程的状态变化和当前线程转储，根据线程转储图形化地显示线程锁定状态，分析线程死锁情况。</li> <li>分析估计线程阻塞对象和阻塞时间。</li> </ul> </li> <li><b>3. 方法采样分析结果</b> <ul style="list-style-type: none"> <li>分析 Java 及 native 代码中热点函数 CPU Cycles 的占比及定位。</li> <li>支持通过火焰图查看热点函数及其调用栈。</li> </ul> </li> <li><b>4. 内存分析结果</b> <ul style="list-style-type: none"> <li>Java 对象在 Heap 中的分配情况，快速定位消耗 Heap 最多或数量分配最多的对象，发现潜在问题；通过堆栈追踪定位潜在的内存问题。</li> <li>通过对存留周期长的 Java 对象进行采样分析，发现潜在的堆内存泄露点，定位潜在堆内存泄露问题。</li> </ul> </li> <li><b>5. GC 分析结果</b> <p>用户查看 Java 的 GC 配置，Heap 大小变化以及 GC 事件发生情况。通过观察 Heap 变化，GC 活动频率和暂停时间，分析和调整当前的 GC 策略。</p> </li> <li><b>6. IO 分析结果</b> <p>用户可分析目标 Java 应用中关于文件读写、Socket 流量使用情况，以发现 IO 使用瓶颈；分析应用对文件的读写统计信息：读写路径、读 / 写频次、读 / 写速率、读 / 写总量、堆栈跟踪（可配）及随时间变化图。</p> </li> </ol>

## 在线分析

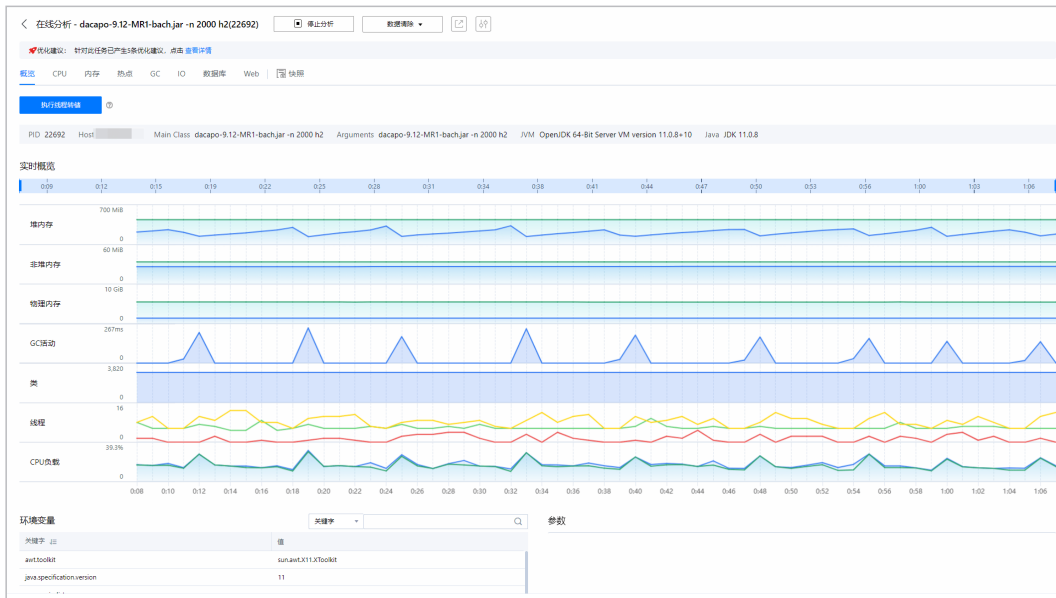
在线分析包含对于目标 JVM 和 Java 程序的双重分析。

**步骤 1** 选择 Java 进程，如图 3-61 所示。



**步骤 2** 查看 Java 性能分析概览，如图 3-62 所示。

图 3-62 概览



---- 结束

## 采样分析

通过采样的方式，收集 JVM 的内部活动 / 性能事件，通过录制及回放的方式来进行离线分析。

**步骤 1** 新建采样分析记录，如图 3-63 所示。

图 3-63 新建采样分析记录

### 新建采样分析记录

记录方式  指定记录时长  滚动录制方式

\* 采样时长 (s)  (1 ~ 300)

方法采样

\* Java方法采样间隔 (ms)  (1 ~ 1000)

\* Native方法采样间隔 (ms)  (1 ~ 1000)

线程转储

\* 转储间隔 (s)  (1 ~ 300)

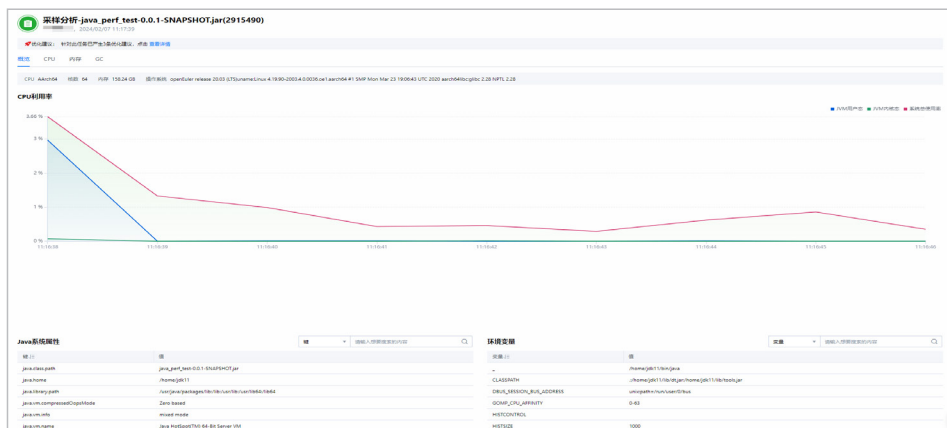
文件IO采样

Socket IO采样

老年代对象采样

**步骤 2** 查看采样分析结果，如图 3-64 所示。

图 3-64 查看采样分析结果



--- 结束

### 3.1.5.4 系统诊断工具

#### 系统诊断工具简介

系统诊断是针对基于鲲鹏的服务器的性能分析工具，通过分析系统运行指标，识别异常点，例如：内存泄漏、内存越界、网络丢包等，并给出优化建议，帮助用户识别出源代码中内存使用的问题点，提升程序的可靠性。

图 3-65 系统诊断工具



表 3-3 任务描述

任务分类	描述
内存使用诊断	分析应用程序存在的内存泄漏点（包括内存未释放和异常释放），得出具体的泄漏信息，并支持关联出调用栈信息和源码。
内存越界诊断	分析应用程序的内存异常访问点，给出异常访问类型和内存访问信息，并支持关联出调用栈和源码。
网络 IO 诊断	压测网络，获得网络最大能力，为网络 IO 性能优化提供基础参考数据；诊断网络，定位网络疑难问题，解决因网络配置和异常而导致的网络 IO 性能问题。具体包括：网络拨测、丢包诊断（RDMA 暂不支持）、网络抓包（RDMA 暂不支持）、系统负载监控。实现网络数据流统计功能，对 IPv4/IPv6 协议簇中 UDP 和 TCP 协议数据流、RDMA RoCEv2、IB 数据流进行分析，统计不同阶段数据流在不同处理核上执行的情况。
存储 IO 诊断	压测存储 IO，获得存储设备最大能力，为存储 IO 性能优化提供基础参考数据。支持存储 IO 压测，获得存储 IO 最大能力，包括：吞吐量、IOPS、时延。

## 内存诊断

步骤 1 创建内存诊断任务，如图 3-66、图 3-67 所示。

图 3-66 内存使用任务

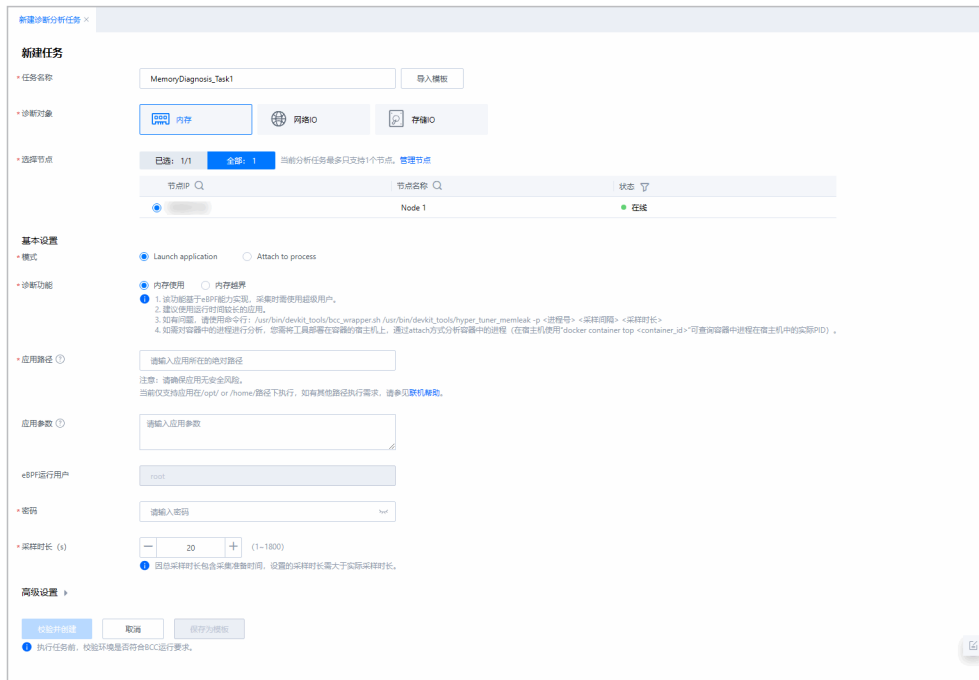
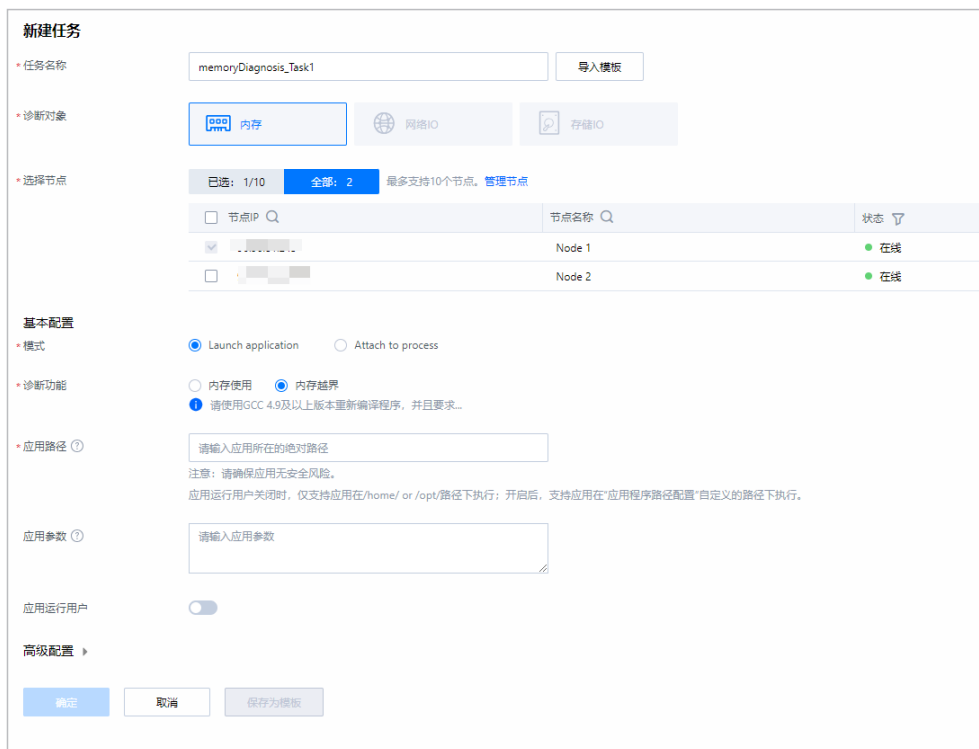


图 3-67 内存越界任务



步骤 2 查看内存诊断任务结果，如图 3-68、图 3-69 所示。

图 3-68 内存使用任务结果

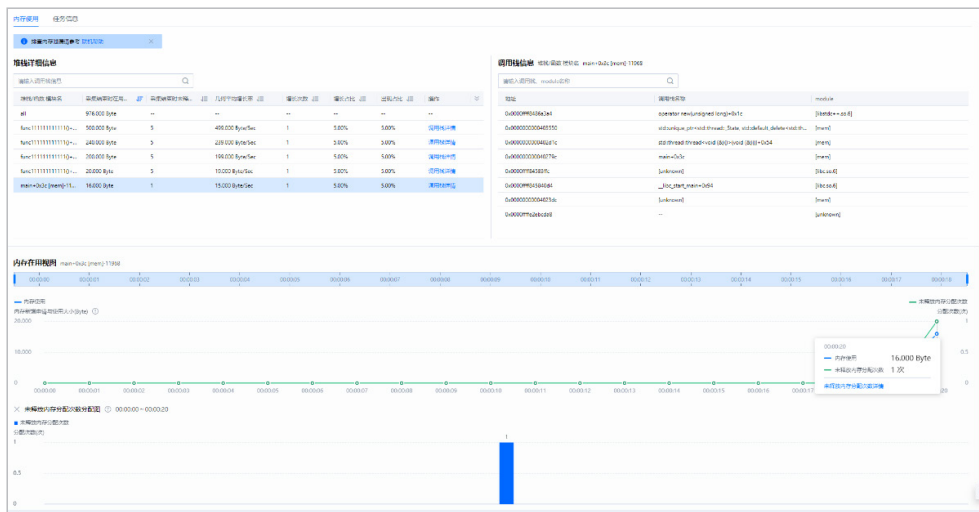
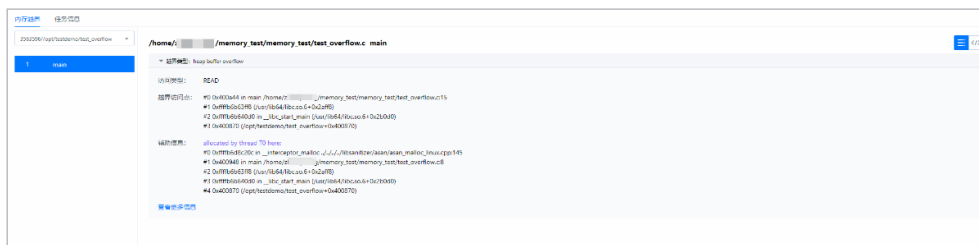


图 3-69 内存越界任务结果



---- 结束

## 网络 IO 诊断

步骤 1 创建网络 IO 诊断任务，如图 3-70 所示。

图 3-70 网络 IO 诊断任务

**新建任务**

任务名称: netioDiagnostic\_Task14 [ 导入模板 ]

诊断对象: [ 内存 ] [ **网络IO** ] [ 存储IO ]

诊断类型:  普通诊断  RDMA诊断  
支持对以太网卡进行网络探测、网络负载监控、丢包和抓包诊断，为网络优化提供参考数据并帮助定位网络疑难问题。

诊断功能:  网络探测  丢包诊断 [ 网络抓包 ] [ 网络负载监控 ]

**网络探测参数**

\* 探测场景: 连通性探测  
TCP/UDP探测任务会消耗大量网络带宽，且需要两个节点；TCP侧重带宽、数据重传；UDP侧重时延抖动。

\* IP协议类型:  IPv4  IPv6

\* 节点信息

\* 任务节点: [ 90. ]

源IP: [ 90. ]

\* 目标服务器IP: [ 90. ]

探测高级参数 ^

任务时间:  立即执行  预约定时启动

[ 确认 ] [ 取消 ] [ 保存为模板 ]

步骤 2 查看网络 IO 诊断任务分析结果，如图 3-71 所示。

图 3-71 网络 IO 诊断任务分析结果

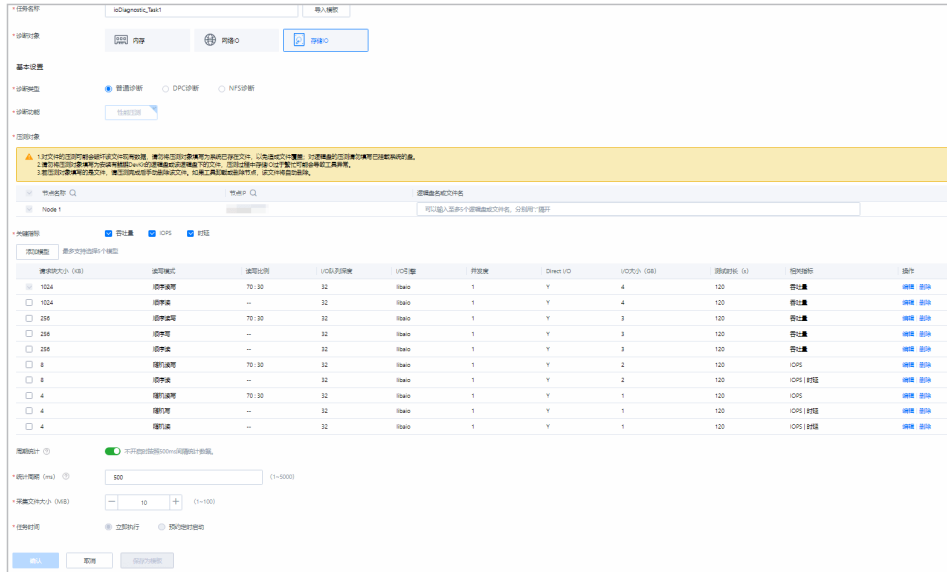


---- 结束

## 存储 IO 诊断

步骤 1 创建存储 IO 诊断任务，如图 3-72 所示。

图 3-72 存储 IO 诊断任务



步骤 2 查看存储 IO 诊断任务分析结果，如图 3-73 所示。

图 3-73 存储 IO 诊断任务分析结果

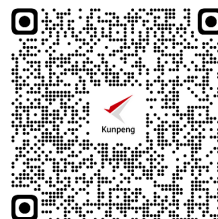


---- 结束

# 3.2 流水线阶段

DevKit 原生开发插件支持 1 小时接入两大主流（Jenkins、GitLab）CI/CD 流水线，用户可以在流水线中高效使用鲲鹏迁移扫描、亲和分析、编译、测试、调优等能力，提升鲲鹏流水线搭建及版本发布效率的同时，性能开箱即优。

鲲鹏 DevKit 快速接入流水线的详细介绍与操作指导请参见[鲲鹏社区](#) → [鲲鹏原生开发](#) → [查看用户指南](#) → [DevKit Pipeline 介绍](#)，详情可扫描右方二维码。

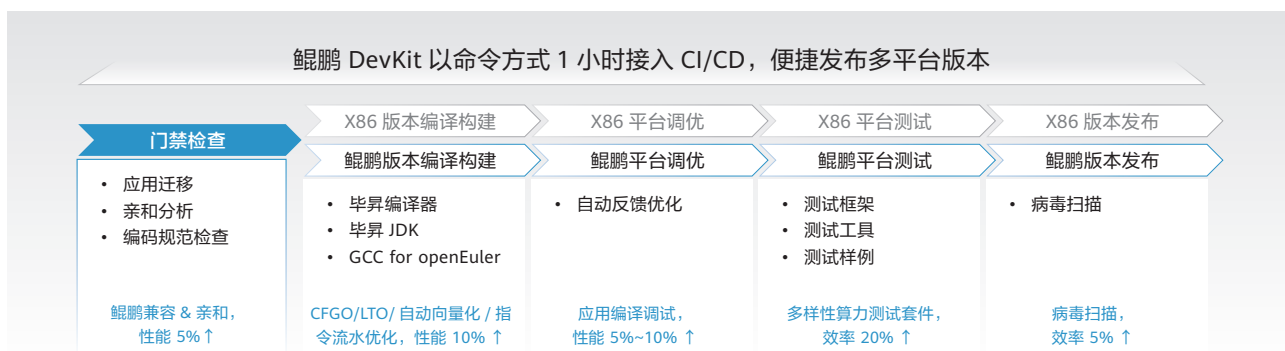


## 3.2.1 门禁检查

### 3.2.1.1 门禁检查介绍

鲲鹏原生开发在流水线门禁检查阶段提供了 DevKit 应用迁移工具、亲和分析工具和编码规范检查，可拦截不适配鲲鹏服务器的代码、提供鲲鹏亲和代码优化建议和编码规范问题提示，把问题拦截在早期阶段，提升鲲鹏原生开发效率。

图 3-74 门禁检查



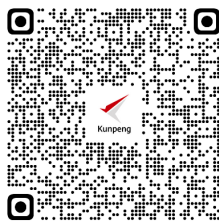
### 3.2.1.2 应用迁移命令行

应用迁移工具属于鲲鹏 DevKit 的扩展工具之一，支持在 x86 和鲲鹏平台运行，提供 VS Code 插件、Web、命令行等多种使用形态，其中命令行工具支持以下功能：

- » 软件迁移评估：自动扫描并分析软件包（非源码包）、已安装的软件，提供可迁移性评估报告。
- » 源码迁移：检查分析 C/C++/ASM/Fortran/ 解释型语言等源码文件，定位出需要迁移的代码并给出迁移指导，支持迁移编辑及一键代码替换功能。

用户可在 CI/CD 流水线中集成应用迁移工具命令行，实现不适配鲲鹏平台代码的门禁拦截能力。

应用迁移命令行详细操作步骤请参见《Kunpeng DevKit 用户指南（命令行工作模式）》中的[应用迁移](#)，详情可扫描下方二维码。



## 软件迁移评估

软件迁移评估帮助用户分析用户 x86 环境下软件包安装路径中的 SO 库文件，并检查这些文件与鲲鹏平台的兼容性。

命令格式：

```
devkit porting pkg-mig {-i INPUT_PATH | --input INPUT_PATH} [-o OUTPUT_PATH | --output OUTPUT_PATH] [-r {all,json,html,csv} | --report-type {all,json,html,csv}] [-l {0,1,2,3} | --log-level {0,1,2,3}] [--set-timeout TIMEOUT] [-t,--target-os target-os]
```

## 源码迁移

源码迁移功能分析用户 C/C++/ASM/Fortran/Go/ 解释型语言软件的可迁移性。

命令格式：

```
devkit porting src-mig {-i INPUT_PATH | --input INPUT_PATH} {-c 'command' | --cmd 'command'} [-s {c|c++|asm|fortran|go| interpreted}* | --source-type {c|c++|asm|fortran|go| interpreted}*] [-t targe_os | --target-os targe_os] [-p compiler_version | --compiler compiler_version] [-f fortran-compiler-version | --fortran-compiler fortran-compiler-version] [-o OUTPUT_PATH | --output OUTPUT_PATH] [-b {make,cmake,automake,go} | --build-tool {make,cmake,automake,go}] [-r {all,json,html,csv} | --report-type {all,json,html,csv}] [-l {0,1,2,3} | --log-level {0,1,2,3}] [--set-timeout TIMEOUT] [--ignore ignore_file] [--macro MACRO] [--keep-going {True,False}]
```



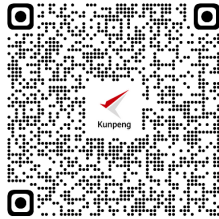
### 3.2.1.3 亲和分析命令行

亲和分析工具属于鲲鹏 DevKit 的扩展工具之一，支持鲲鹏平台软件代码质量检查，提升代码质量、优化访存性能，提供 VS Code 插件、Web、命令行等多种使用形态，其中命令行工具支持以下功能：

- » 64 位运行模式检查：使用命令通过终端工具进行运行模式检查。
- » 字节对齐检查：检查源码中结构体变量的字节对齐情况。
- » 内存一致性检查：检查源码中存在的内存一致性问题。
- » 向量化检查：对可向量化片段进行检查。
- » 矩阵化检查：对可矩阵化片段进行检查。
- » 构建亲和：对构建亲和进行检查。
- » 缓存行对齐检查：对 C/C++ 源码中结构体变量进行 128 字节对齐检查。
- » BC 文件生成：通过扫描源码文件生成对应的 BC 文件。

用户可在 CI/CD 流水线中集成亲和分析工具命令行，实现鲲鹏亲和代码的优化建议。

亲和分析命令行详细操作步骤请参见《Kunpeng DevKit 用户指南（命令行工作模式）》中的[亲和分析](#)，详情可扫描下方二维码。



## 64 位运行模式检查

4 位运行模式检查就是将 GCC4.8.5 ~ GCC10.3.0 版本原 32 位的应用迁移到 64 位平台上，进行迁移检查并给出修改建议。

命令格式：

```
devkit advisor run-mode {-i INPUT_PATH | --input INPUT_PATH} [-o OUTPUT_PATH | --output OUTPUT_PATH] [-l {0,1,2,3} | --log-level {0,1,2,3}] [-r {all,json,html,csv} | --report-type {all,json,html,csv}] [--set-timeout TIMEOUT]
```

## 字节对齐检查

字节对齐检查就是在需要考虑字节对齐时，检查源码中结构体类型变量的字节对齐情况，提升内存访问效率。

命令格式：

```
devkit advisor addr-align {-i INPUT_PATH | --input INPUT_PATH} {-c COMMAND | --cmd COMMAND} [-o OUTPUT_PATH |
--output OUTPUT_PATH]
[-b {make,cmake, automake} | --build-tool {make,cmake, automake}] [-r {all,json,html,csv} | --report-type
{all,json,html,csv}]
[-l {0,1,2,3} | --log-level {0,1,2,3}] [--set-timeout TIMEOUT]
```

## 内存一致性检查

内存一致性检查就是检查源码迁移在鲲鹏平台运行时可能存在的内存一致性问题，并提供插入内存屏障的建议。

命令功能：

```
devkit advisor mm-check {-i INPUT_PATH | --input INPUT_PATH} [-f BC_PATH | --bc-file BC_PATH] [--autofix {true,false}]
[--autofix-dir AUTOFIX_OUTPUT_PATH] [-o OUTPUT_PATH | --output OUTPUT_PATH] [-r {all,json,html,csv} | --report-type
{all,json,html,csv}] [-l {0,1,2,3} | --log-level {0,1,2,3}]
[--set-timeout TIMEOUT]
```

## 向量化检查

向量化检查功能用于对可向量化片段进行检查，并提供向量化修改建议，更好地发挥芯片性能。

命令格式：

```
devkit advisor vec-check {-i INPUT_PATH | --input INPUT_PATH} [-f BC_PATH | --bc-file BC_PATH] [-c COMMAND | --cmd
COMMAND] [-p {clang,gcc} | --compiler {clang,gcc}] [-o OUTPUT_PATH | --output OUTPUT_PATH] [-r {all,json,html,csv}
| --report-type {all,json,html,csv}] [-l {0,1,2,3} | --log-level {0,1,2,3}] [--set-timeout TIMEOUT] [--sve-enable
{true,false}]
```

## 矩阵化检查

矩阵化检查功能用于对可矩阵化片段进行检查，并提供矩阵化修改建议，更好地发挥芯片性能。

命令功能：

```
devkit advisor matrix-check {-i INPUT_PATH | --input INPUT_PATH} [-s scan_file_path | --scan-dir scan_file_path]
[(-b {make,cmake} | --build-tool {make,cmake}) & (-c COMMAND | --cmd COMMAND)] | [-j COMPILER_JSON_PATH | --compile-
command-json COMPILER_JSON_PATH] [-o OUTPUT_PATH | --output OUTPUT_PATH] [-r {all,json,html,csv} | --report-type
{all,json,html,csv}] [-l {0,1,2,3} | --log-level {0,1,2,3}] [-p {sme,domain} | --optimization {sme,domain}] [-m
{compute,memory_access,communication} | --module {compute,memory_access,communication}] [--set-timeout TIMEOUT]
```

## 构建亲和

构建亲和功能用于分析 makefile、CMakeLists.txt 中可以替换鲲鹏加速库的内容，并提供替换建议和功能修复。

命令功能：

```
devkit advisor affi-check {-i INPUT_PATH | --input INPUT_PATH} {-c COMMAND | --cmd COMMAND} [-o OUTPUT_PATH | --output OUTPUT_PATH] [-b {make,cmake} | --build-tool {make,cmake}] [-r {all,json,html,csv} | --report-type {all,json,html,csv}] [-l {0,1,2,3} | --log-level {0,1,2,3}] [--set-timeout TIMEOUT]
```

## 缓存行对齐检查

缓存行对齐检查是对 C/C++ 源码中结构体变量进行 128 字节对齐检查，提升访存性能。

命令功能：

```
devkit advisor cacheline {-i INPUT_PATH | --input INPUT_PATH} [-o OUTPUT_PATH | --output OUTPUT_PATH] [-r {all,json,html,csv} | --report-type {all,json,html,csv}] [-l {0,1,2,3} | --log-level {0,1,2,3}] [--set-timeout TIMEOUT]
```

## BC 文件生成

BC 文件用于内存一致性检查和向量化检查，简化 BC 文件生成过程。

命令功能：

```
devkit advisor bc-gen {-i INPUT_PATH | --input INPUT_PATH} {-c COMMAND | --cmd COMMAND} [-o OUTPUT_PATH | --output OUTPUT_PATH] [-l {0,1,2,3} | --log-level {0,1,2,3}] [--set-timeout TIMEOUT]
```

### 📍 3.2.1.4 编码规范检查

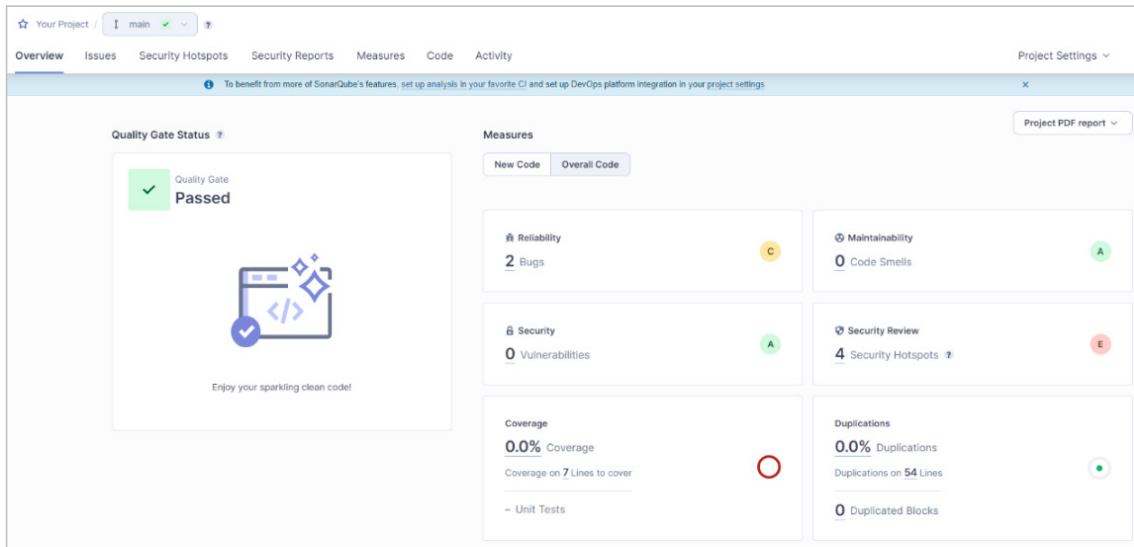
鲲鹏原生开发流水线使用 SonarQube 进行编码规范检查。SonarQube 是一个管理代码质量的开放平台，涵盖了架构设计、注释、编码规范、潜在缺陷、代码复杂度、重复代码等多个维度的质量分析。帮助检查代码缺陷、改善代码质量以及提高开发速度，可以支持 Java、C、C++、JavaScript 等多种编程语言的代码质量管理与检测。

包含七个代码质量检查维度：

- » 复杂度分布 (complexity)：代码复杂度过高将难以理解。
- » 重复代码 (duplications)：程序中包含大量复制、粘贴的代码而导致代码臃肿，SonarQube 可以展示源码中重复严重的地方。
- » 单元测试统计 (unit tests)：统计并展示单元测试覆盖率，开发或测试可以清楚测试代码的覆盖情况。

- » 代码规则检查 (coding rules) : 通过 Findbugs、PMD、CheckStyle 等检查代码是否符合规范。
- » 注释率 (comments) : 若代码注释过少, 当有人员变动后, 其他人比较难以接手; 若代码注释过多, 又不利于阅读。
- » 潜在的 Bug (potential bugs) : 通过 Findbugs、PMD、CheckStyle 等检测潜在的 bug。
- » 结构与设计 (architecture & design) : 找出循环, 展示包与包、类与类之间的依赖、检查程序之间耦合度。

图 3-75 代码缺陷检查



## 3.2.2 编译构建

鲲鹏原生开发在流水线编译构建阶段提供了毕昇编译器、毕昇JDK、GCC for openEuler。通过上述鲲鹏亲和编译器, 在不修改用户应用源码的情况下, 提供鲲鹏平台的应用性能提升。

详细内容请参见 3.1.3 编译。

图 3-76 编译构建



### 3.2.3 调优

#### 3.2.3.1 调优介绍

鲲鹏原生开发在流水线调优阶段提供了自动反馈优化功能，通过流水线自动触发，实现应用编译优化。

图 3-77 调优



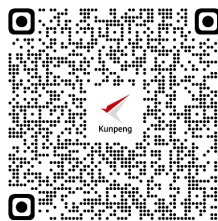
#### 3.2.3.2 自动反馈优化

PGO ( Profile Guided Optimization ) 是一种编译器优化技术，通过编译器插桩收集程序运行时信息进行优化决策。编译器根据这些运行时信息指导各种编译优化技术进行更准确的优化决策，生成更优的目标程序。

A-FOT 是一款用于提升编译器 GCC for openEuler 自动反馈优化特性的工具，支持以下三种模式：

- » AutoFDO: PGO 的简化部署版，使用 perf 替代插桩获取程序运行 profile，受益优化点包括矢量化、循环展开、循环剥离等优化。
- » AutoPrefetch: 增强版预取优化，根据 Dcache 访问、存取指令 Cache Miss 率，获取数据访问优化代码块并进行预取优化，同时优化预取提前量，建议和 AutoFDO 共同使用。
- » AutoBOLT: 链接后二进制优化，对控制流复杂的程序具有显著的优化效果，主要优化包括 BB 重排、函数重排、冷热分区等优化，与 AutoFDO、AutoPrefetch 部分冲突。

A-FOT 安装部署指导请参见《鲲鹏原生开发 用户指南》中的[安装 A-FOT 工具](#)，详情可扫描下方二维码。

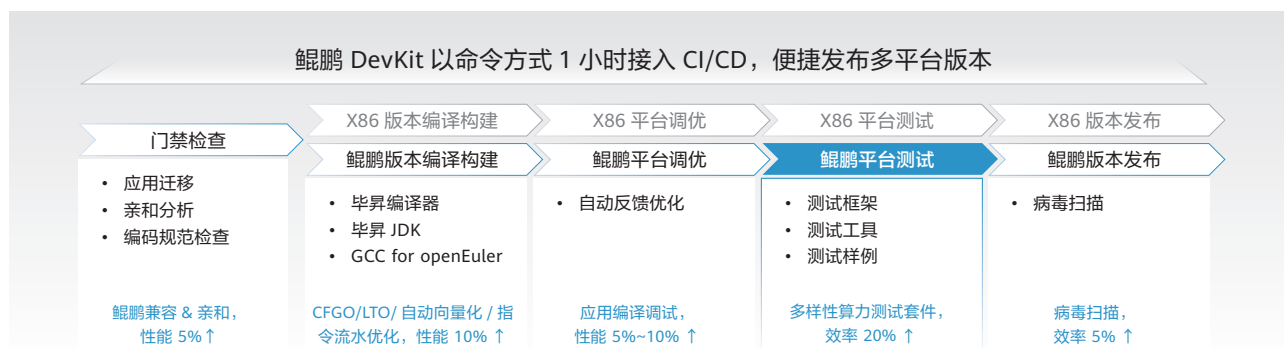


## 3.2.4 测试

### 3.2.4.1 测试介绍

鲲鹏原生开发在流水线测试阶段提供了兼容性测试工具和 Java 性能测试工具，在鲲鹏服务器上对应用进行全方位测试。

图 3-78 测试



### 3.2.4.2 兼容性测试工具

鲲鹏原生开发提供了一系列兼容测试用例集，实现了鲲鹏兼容测试自动化，保证应用实现鲲鹏兼容。

测试用例集如表 3-4 所示。

表 3-4 测试用例集说明

任务分类	描述
兼容性测试	通过待测试应用软件在鲲鹏环境启动前后资源波动异常检测、验证应用软件启动和停止，自动检测应用软件在鲲鹏平台上的可运行性、兼容性问题。
可靠性测试	通过待测试应用软件在稳定运行期间的系统资源内存的波动异常检测、在异常终止测试场景检测应用运行，自动评估应用软件在鲲鹏平台上的稳定性和可靠性。
安全测试	通过待测试应用软件在运行情况下的端口扫描、病毒扫描和漏洞扫描，自动发现应用软件可能存在的安全风险。

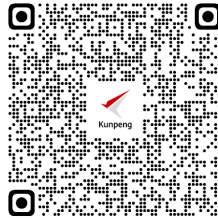
鲲鹏原生开发兼容性测试工具安装部署详细内容请参见《鲲鹏原生开发 用户指南》中的[单独部署兼容性测试工具](#)，详情可扫描下方二维码。



### 3.2.4.3 Java 性能测试工具

鲲鹏原生开发在流水线测试阶段提供了 Java 性能测试工具，结合 Jmeter，对目标程序进行压测，采集目标进程的性能数据，生成压测报告。

鲲鹏原生开发 Java 性能测试工具安装部署详细内容请参见《鲲鹏原生开发 用户指南》中的[安装工具到执行机](#)，详情可扫描下方二维码。



## 3.2.5 版本发布

鲲鹏原生开发在版本发布阶段提供了病毒扫描能力，实现了版本安全合规扫描自动化，版本发布效率提升。

图 3-79 版本发布



鲲鹏原生开发病毒扫描使用开源病毒扫描工具 ClamAV，ClamAV 可检测数百万种病毒、蠕虫、木马和其他恶意软件，包括 Microsoft Office 宏病毒、移动恶意软件和其他威胁。

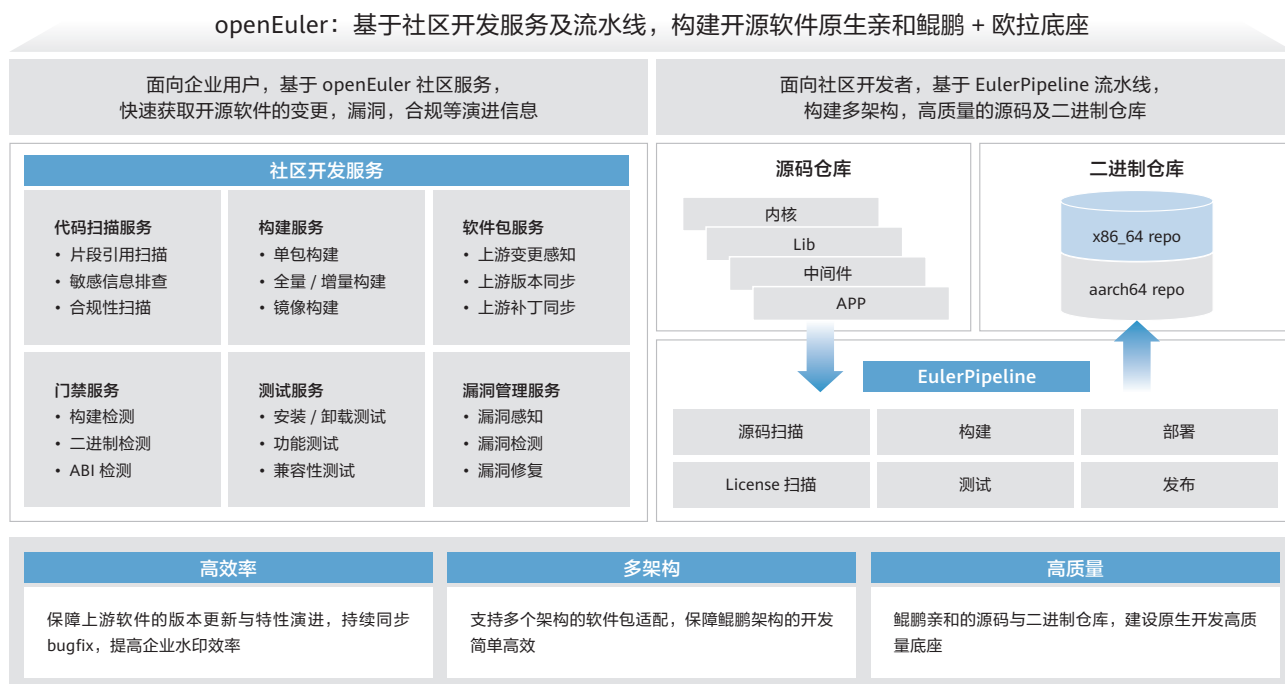
# 3.2 OS底座 (openEuler系操作系统)

## 3.3.1 openEuler 简介

- » openEuler 是一个面向数字基础设施的 AI 原生操作系统，支持服务器、云计算、边缘计算、嵌入式等应用场景，支持多样性计算，致力于提供安全、稳定、易用的操作系统。
- » openEuler 社区及其他第三方开发者共同提供了丰富易用的软件包，在最新的版本中，软件包总数已超过 3.5 万。openEuler 社区版本使用者可以根据自己的需求配置不同的软件仓库。
- » openEuler 从用户场景出发，回溯梳理相应的软件依赖关系，理清所有软件包的上游社区地址、源码和上游对应验证。完成构建验证、分发、实现生命周期管理，提供可靠开源软件供应链。

目前，openEuler 社区积累超过 1800+ 的企业和合作伙伴，覆盖了全产业链，发展了近万名社区贡献者，国内外主流操作系统厂商均推出基于欧拉的商业发行版。

图 3-80 openEuler 简介



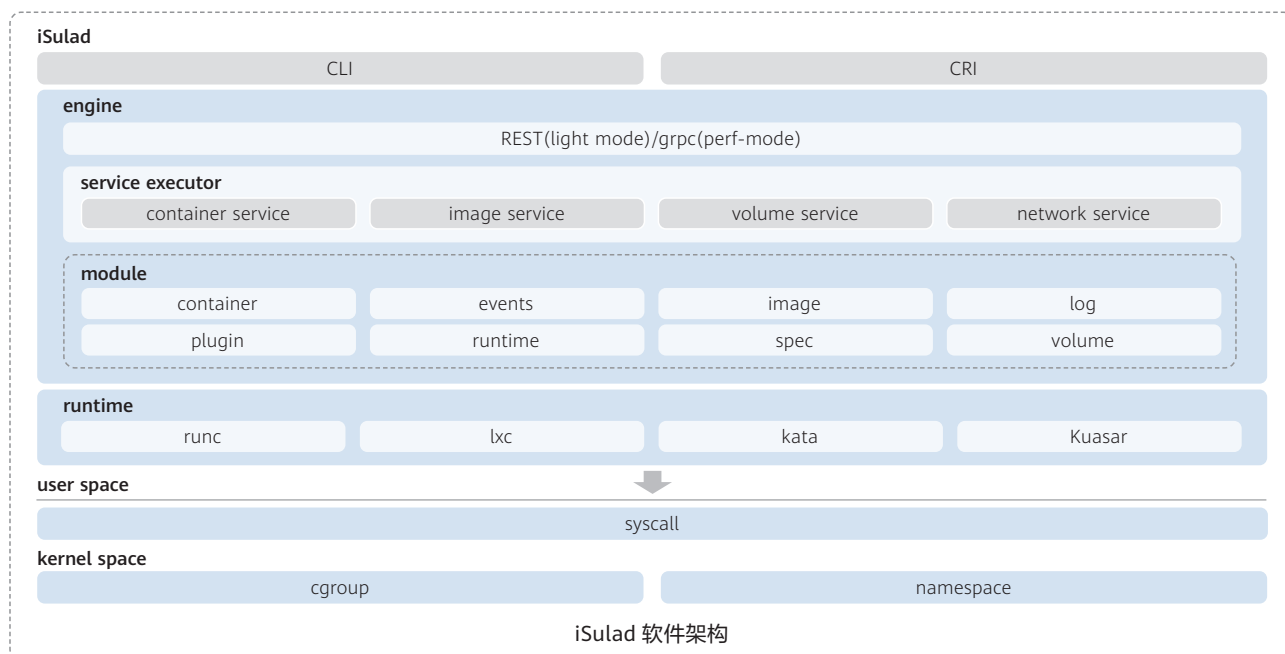
获取最新的 openEuler 版本和详细的用户指南，请参见 [openEuler 社区](https://www.openeuler.org/zh/) ( <https://www.openeuler.org/zh/> )。

## 3.3.2 openEuler 基础能力

### 3.3.2.1 iSulad 轻量级容器引擎

- » iSulad 是一个由 C/C++ 编写实现的轻量级容器引擎，具有轻、灵、巧、快的特点，不受硬件规格和架构限制，底噪开销更小，可应用的领域更为广泛。
- » iSulad 是 openEuler 提供的新的容器引擎，其统一的架构设计能够满足 CT 和 IT 领域的不同需求。相比 Golang 编写的 Docker，iSulad 资源占用更少，容器启动更快，可应用范围更广。
- » iSulad 容器引擎提供了与 Docker 类似的命令行，方便用户操作使用。其北向支持 CRI 接口，可以对接 Kubernetes，用户可以使用 iSulad 作为底座，通过 Kubernetes 进行容器的编排调度。iSulad 南向支持 OCIruntime 标准，能够灵活对接 runc、lxc、kata、kuasar 等多种容器运行时，兼容容器生态。

图 3-81 iSulad



iSulad 的核心能力包括容器服务，镜像服务、卷服务以及网络服务。

- » 容器服务：用来负责容器生命周期的管理。
- » 镜像服务：负责提供对容器镜像的操作。

iSulad 支持符合 OCIimage 标准的镜像格式，保证 iSulad 能够支持业界主流镜像。此外，iSulad 还支持用于系统容器场景的 externalrootfs 以及嵌入式场景的 embedded 镜像格式。

- » 卷服务：为用户提供容器数据卷管理的能力。
- » 网络服务：可以与符合 CNI 标准的网络插件一起，为容器提供网络能力。

iSulad 作为一款通用容器引擎，除了支持运行普通容器之外，还支持运行系统容器与安全容器。

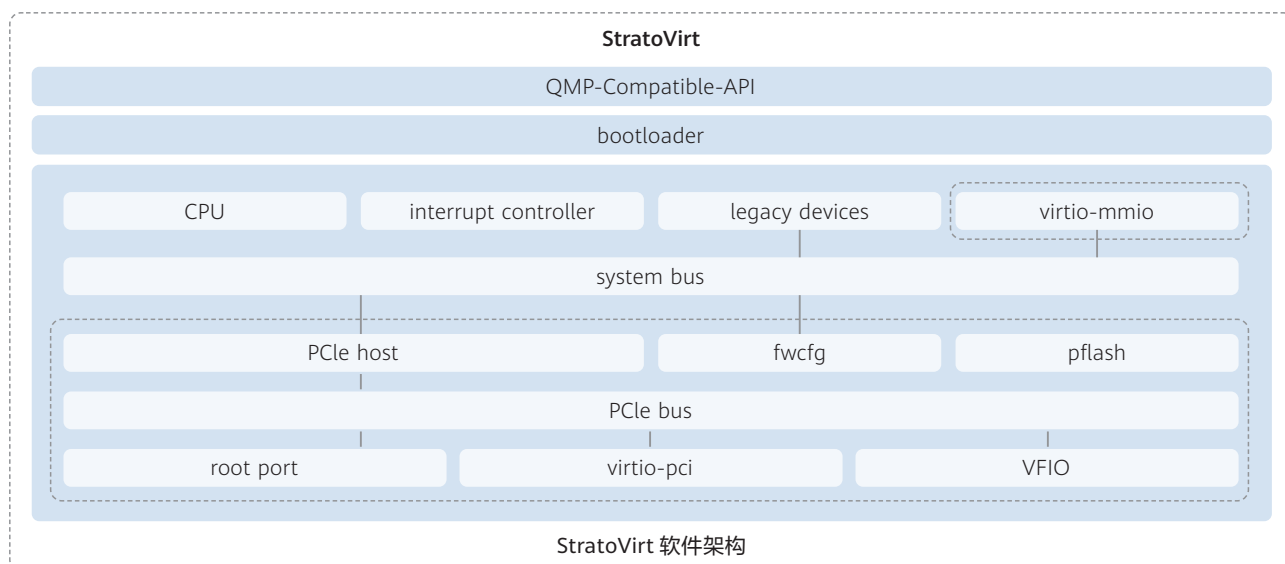
- » 普通容器：传统的应用容器。
- » 系统容器：在普通容器的基础上进行功能扩展，相比于普通容器，系统容器具备 systemd 管理服务的能力，支持在容器运行时动态添加 / 释放磁盘设备、网卡、路由以及卷。系统容器主要应用在重计算、高性能、大并发的场景下，可以解决重型应用和业务云化的问题。
- » 安全容器：安全容器是虚拟化技术和容器技术的结合，相比于普通容器共用同一台宿主机内核存在的安全隐患，安全容器通过虚拟化层实现容器间的强隔离，每个安全容器都有一个自己单独的内核和轻量级虚拟机运行环境，保证同一个宿主机上不同安全容器的运行互相不受影响。

### 🔗 3.3.2.2 StratoVirt

StratoVirt 是一种基于 Linux 内核虚拟化（KVM）的开源轻量级虚拟化技术，在保持传统虚拟化的隔离能力和安全能力的同时，降低了内存资源消耗，提高了虚拟机启动速度。StratoVirt 可以应用于微服务或函数计算等 Serverless 场景，保留了相应接口和设计，用于快速导入更多特性，直至支持通用虚拟化。

StratoVirt 的核心架构如图 3-82 所示，从上到下分为三层。

图 3-82 StratoVirt



- » 外部 API：StratoVirt 使用 QMP 协议与外部系统通信，兼容 OCI，同时支持对接 libvirt。
- » bootloader：轻量化场景下使用简单的 bootloader 加载内核镜像实现快速启动，而不像传统的繁琐的 BIOS 和 Grub 引导方式；在通用虚拟化场景下，支持 UEFI 启动。
- » 模拟主板 microvm：为了提高性能和减少攻击面，StratoVirt 最小化了用户态设备的模拟。模拟实现了 KVM 仿真设备和半虚拟化设备，如 GIC、串行、RTC 和 virtio-mmio 设备。
- » 通用机型：提供 ACPI 表实现 UEFI 启动，支持添加 virtio-pci 以及 VFIO 直通设备等，极大提高虚拟机的 I/O 性能。

StratoVirt 配合 iSulad 容器引擎和 Kubernetes 编排引擎可形成完整的容器解决方案，支持 Serverless 负载高效运行。

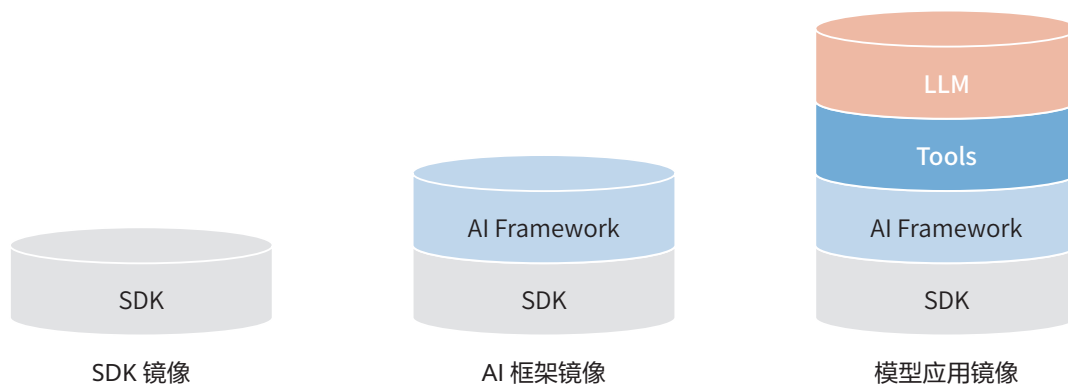
### 🔗 3.3.2.3 openEuler AI 软件栈

openEuler 兼容 NVIDIA、Ascend 等主流算力平台的软件栈，为用户提供高效的开发运行环境。通过将不同 AI 算力平台的软件栈进行容器化封装，即可简化用户部署过程，提供开箱即用的体验。同时，openEuler 也提供丰富的 AI 框架，方便大家快速在 openEuler 上使用 AI 能力。

#### 功能描述：

- » openEuler 已兼容 CANN、CUDA 等硬件 SDK，以及 TensorFlow、PyTorch、MindSpore 等相应的 AI 框架软件，支持 AI 应用在 openEuler 上高效开发与运行。
- » openEuler AI 软件栈容器化封装优化环境部署过程，并面向不同场景提供以下三类容器镜像：

图 3-83 容器镜像



- » SDK 镜像：以 openEuler 为基础镜像，安装相应硬件平台的 SDK，如 Ascend 平台的 CANN 或 NVIDIA 的 CUDA 软件。
- » AI 框架镜像：以 SDK 镜像为基础，安装 AI 框架软件，如 PyTorch 或 TensorFlow。此外，通过此部分镜像也可快速搭建 AI 分布式场景，如 Ray 等 AI 分布式框架。
- » 模型应用镜像：在 AI 框架镜像的基础上，包含完整的工具链和模型应用。

相关使用方式请参见《openEuler AI 容器镜像用户指南》，详情可扫描下方二维码。



## 应用场景

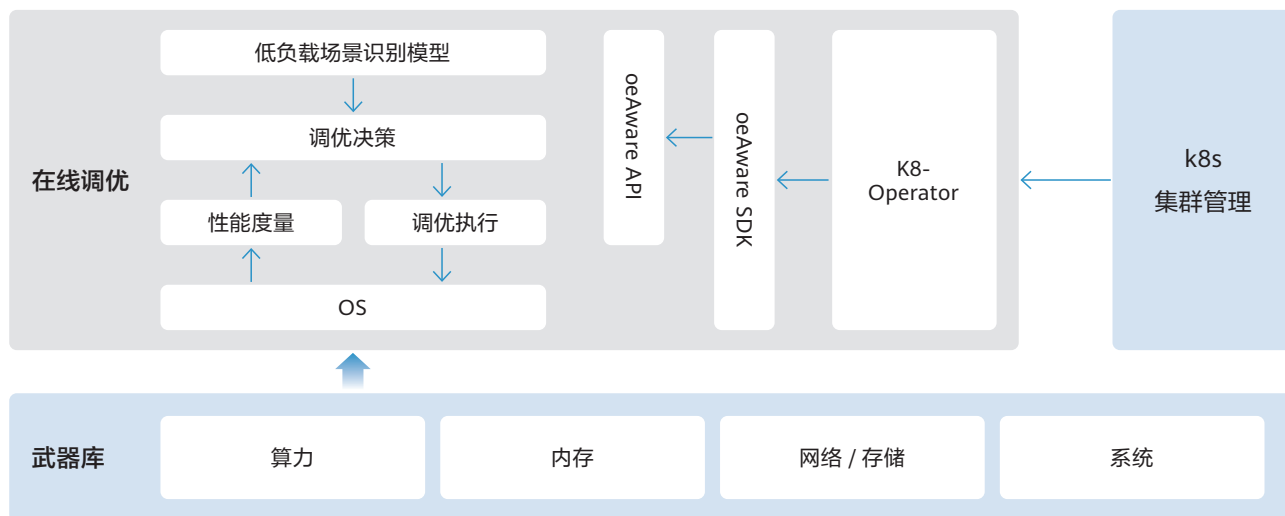
openEuler 使能 AI，向用户提供更多 OS 选择。基于 openEuler 的 AI 容器镜像可以解决开发运行环境部署门槛高的问题，用户根据自身需求选择对应的容器镜像即可一键部署，三类容器镜像的应用场景如下：

- » SDK 镜像：提供对应硬件的计算加速工具包和开发环境，用户可进行 Ascend CANN 或 NVIDIA CUDA 等应用的开发和调试。同时，可在该类容器中运行高性能计算任务，例如大规模数据处理、并行计算等。
- » AI 框架镜像：用户可直接在该类容器中进行 AI 模型开发、训练及推理等任务。
- » 模型应用镜像：已预置完整的 AI 软件栈和特定的模型，用户可根据自身需求选择相应的模型应用镜像来开展模型推理或微调任务。

### 🔧 3.3.2.4 oeAware 智能调优框架

oeAware 是在 openEuler 上实现低负载采集感知调优的框架，目标是在系统 / 业务运行时动态感知系统行为后智能使能系统调优特性。传统调优特性都以独立运行且静态打开 / 关闭为主，oeAware 将调优拆分为采集、感知和调优三层，每层通过订阅方式关联，各层采用插件式开发尽可能复用。oeAware 支持单机部署以及集群协同方式运行。在调优过程中通过规则 + 启发的方式提升调优效率和场景自适应能力，并对业务场景透明。调优前后通过性能度量检测是否存在负优化。并在问题发生后及时上报问题，第一时间回滚止损。

图 3-84 oeAware 调优框架



#### oeAware 优势 & 价值：

oeAware 通过场景感知使能对应调优，支持调优推荐、一键使能以及动态回滚。oeAware 是 openEuler 调优的统一入口，提升调优管理效率，同时对调优一致性以及变更进行管理。

- » 管理面易用性提升：静态调优随着时间延续，不同场景的调优特性会逐渐散乱在系统中，无法高效对调优进行管理和审计。
- » 数据面灵活性提升：一套静态调优配置无法自动适用所有场景；部分调优无法实现静态配置，如复杂场景下的调度策略；场景组合导致调优失效，人工分析成本高。

### 3.3.3 openEuler 工具和社区服务

#### 3.3.3.1 openEuler DevStation 开发者工作站

openEuler DevStation 是 openEuler 首个面向开发者的开发者工作站，预装 VSCODE，大幅提升开发者效率；在之后版本，DevStation 更将推出 oeDeploy，开发插件等功能，并且将打通部署，编码，编译，构建，发布全流程，开发者可以方便的使用 oeDeploy 完成 AI 软件栈，云原生软件栈部署，使用 oeDevPlugin 插件进行一键拉取代码仓，一键使用 AI4C 编译器编译，一键调用 EulerMaker，轻松使用 DevStation 开发者工作站在 openEuler 上进行软件开发。

#### EulerMaker 构建系统

EulerMaker 构建系统是一款软件包构建系统，完成源码到二进制软件包的构建，并支持开发者通过搭积木方式，组装和定制出适合自己需求的场景化 OS。主要提供增量 / 全量构建，分层定制与镜像定制的能力。社区开发者及合作伙伴基于统一构建系统建设自己的用户个人仓、OS 核心仓，定制出适合自己需求的场景化 OS。

图 3-86 EulerMaker 特性



功能描述：

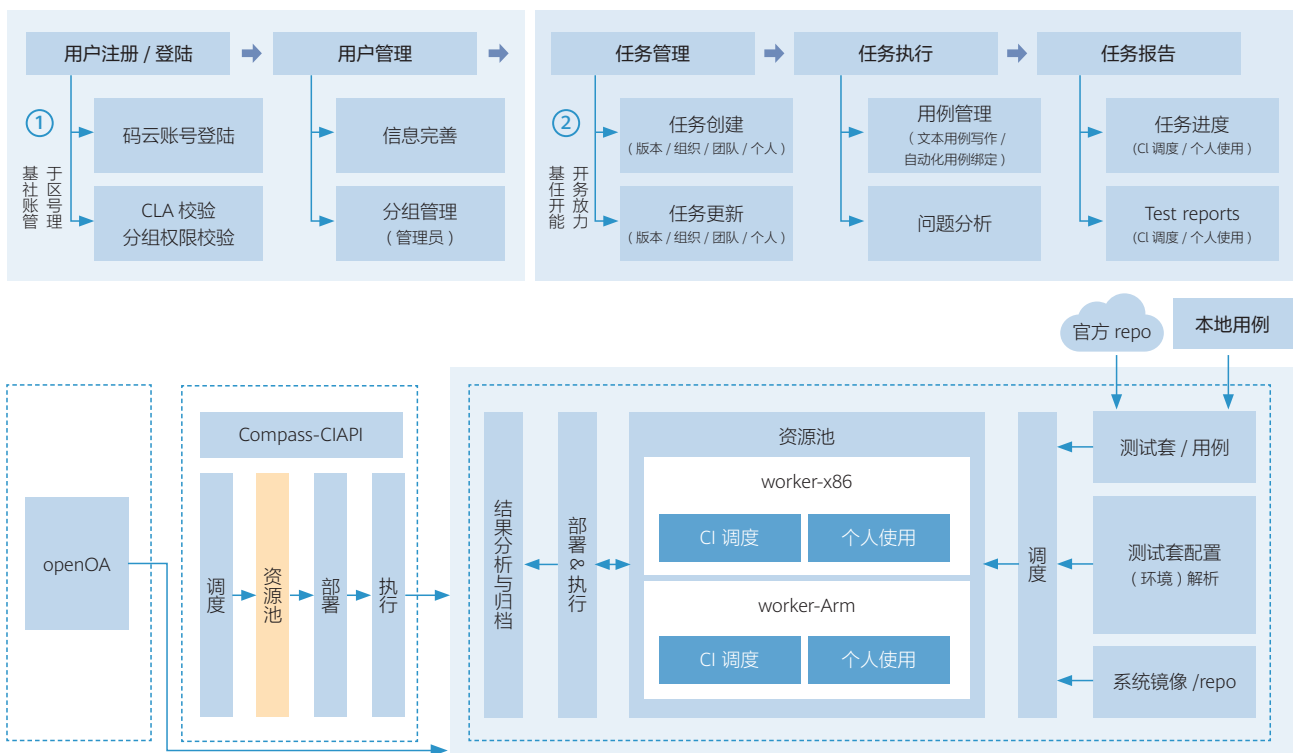
- » 增量 / 全量构建：基于软件包变化并结合软件包依赖关系，分析影响范围，得到待构建软件包列表，按照依赖顺序下发并行构建任务。
- » 构建依赖关系：提供工程软件包构建依赖表，支持筛选及统计软件包依赖及被依赖的软件包内容。
- » 分层定制：支持在构建工程中，基于 spec 或 yaml，叠加配置层模型，实现针对软件包的版本、patch、构建依赖、安装依赖、编译选项及构建流程等内容的定制。

- » 镜像定制: 支持开发者通过配置 repo 源, 生成 iso、qcow2、容器等 OS 镜像, 并支持对镜像进行软件包列表定制。
- » 支持本地任务复现: 通过命令行在本地复现构建任务, 方便定位构建问题。
- » 一键工程创建: 基于 yaml 配置实现一键工程创建, 支持批量加包, 大大简化用户操作。

## EulerTest 测试管理平台

EulerTest 是 openEuler 社区孵化的用以承载社区全流程测试活动的管理平台。EulerTest 核心为 web 端数据中台, 帮助社区版本测试高效运作, 使能社区版本测试可跟踪可追溯。具备支撑资源管理以及自动化测试功能的插件化服务, 支持对接多元测试引擎。

图 3-87 EulerTest 测试管理平台



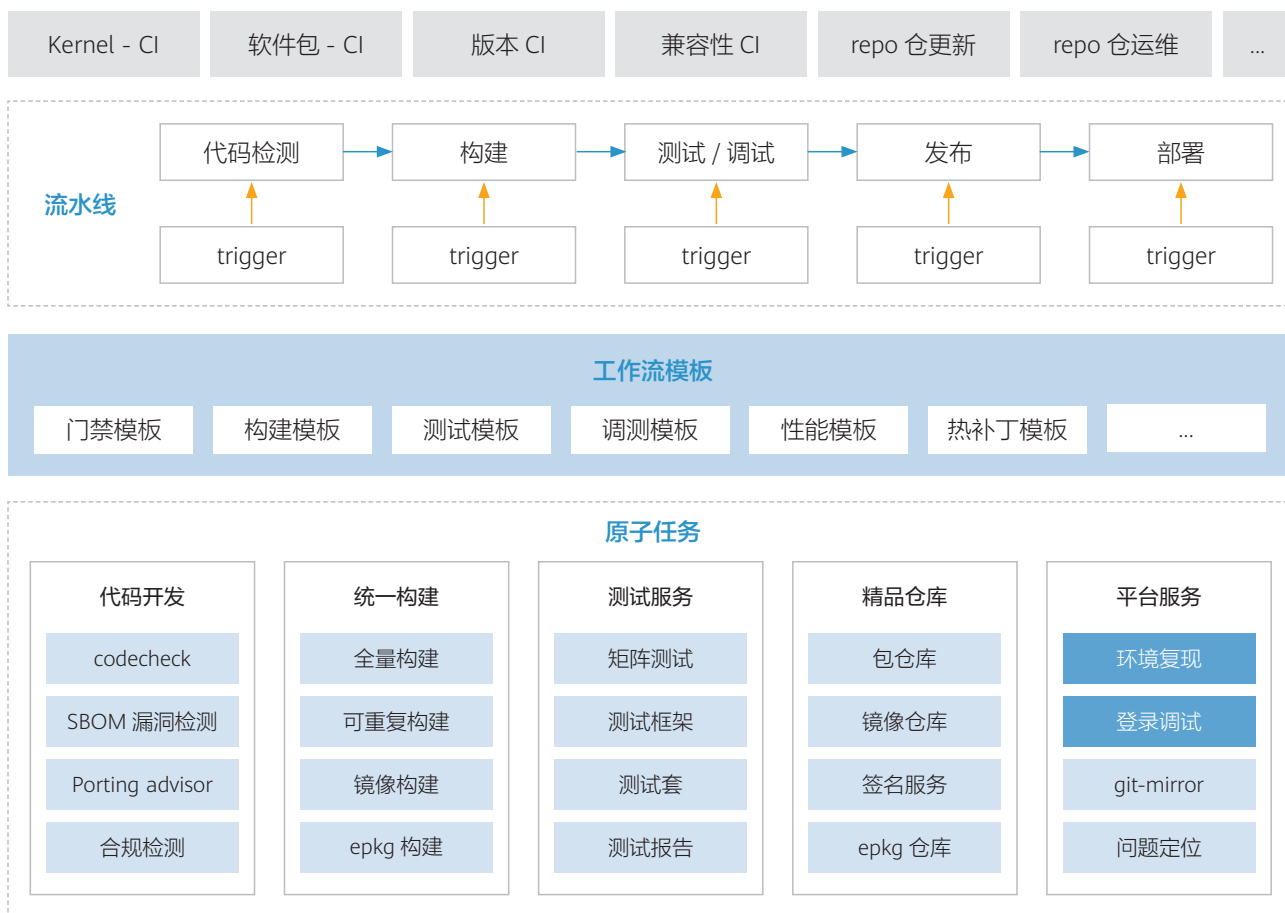
### 功能描述:

- » 支持对物理机进行静态资源管理, 包括资源的密钥修改、占用释放和系统重装。
- » 支持对虚拟机进行动态资源管理, 提供网卡磁盘配置热修改以及 web 控制台等功能。
- » 支持文本用例的数据管理和版本用例基线的制定, 具备用例评审功能。
- » 支持对产品和里程碑进行数据管理, 支持和码云企业仓进行数据同步管理, 提供版本质量看板以支撑质量看护, 使能测试可信。
- » 支持管理测试任务数据, 支持自动化测试触发执行, 以及支持手工测试执行 IT 化管理。具备日志按测试步骤分割和分析标注的功能。
- » 支持自动化从 openQA、Compass-CI 等平台读取执行结果矩阵, 并基于社区既定模板整合为版本测试报告。

## EulerPipeline 流水线

面向社区开发者，基于 EulerPipeline 流水线，构建多架构、高质量的源码及二进制仓库。支持 Package-CI/Code-CI/Repo-CI/ISO-CI/ 单包 test-CI/ 版本 test-CI 6 类模板，实现模板 + 任务按需编排，提供 40+ 原子化 SaaS 服务，简化社区开发者 OS& 软件包开发。支持流水线编排、矩阵测试、一键复现、结对联调。

图 3-88 EulerPipeline



功能描述:

- » 流水线编排: 支持流水线自定义, DSL 语言编排。
- » 矩阵测试: 一次声明, 多种组合并行运行; 提供矩阵总览视图, 多 OS、多架构运行结果一眼掌握。
- » 一键复现: 通过 Web 控制台登入复现环境调测, 同时支持通过自有终端软件利用公钥登入复现环境调测。
- » 结对联调: 流水线服务提供的 web 控制台支持多人共享终端, 使能多用户结对调测。

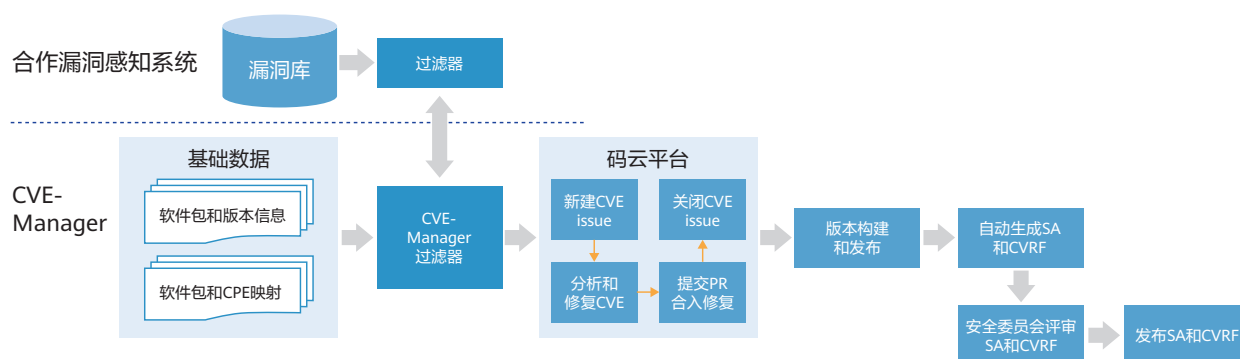
## CVE Manager 漏洞管理

漏洞管理是 openEuler 社区对安全漏洞进行感知、收集、处理以及披露的流程、工具和机制的统称。

从合作漏洞感知系统获取公开漏洞感知信息，再通过机器人在码云平台对应项目软件包仓创建并维护漏洞相关记录，漏洞修复后进入通用版本构建发布以及安全公告发布流程。openEuler 使用 CVS Sv3 进行漏洞评分。

为了保护 openEuler 用户的安全，在进行调查、修复和发布安全公告之前，openEuler 社区不会公开披露、讨论或确认 openEuler 产品的安全问题。

图 3-89 CVE Manager 漏洞管理



功能描述：

- » 发布安全公告：安全公告内容包括该漏洞的技术细节、CVE 编号、CVSS 安全评分、严重性等级以及受到该漏洞影响的版本和修复版本等信息。
- » 安全公告提供邮件订阅功能：社区同时也提供 CVRF 格式的安全公告。

### 3.3.3.2 操作系统迁移

当操作系统生命周期即将 EOM，但是应用软件或者产品的生命周期还未结束时，为了保障业务的连续性、安全性，急需将原操作系统迁移到新操作系统上。企业一方面需要做好操作系统选型（如持续演进、全场景支持、可靠、稳定等），支撑企业持续发展、产品创新、数字化转型；另一方面需要有系统性的迁移方案及工具，保障迁移无忧。针对企业迁移操作系统的需求，openEuler 助力企业简单、平稳、高效进行操作系统迁移。

x2openEuler 是一款将源操作系统迁移到目标操作系统的迁移工具套件，可以对 CentOS, Redhat, Ubuntu, SUSE 进行原地迁移，同时具有批量化原地升级能力，工具提供简单易用的操作界面，可以批量添加待升级节点进行迁移分析，设计迁移方案并对兼容性问题进行迁移适配，最后对已适配的待升级节点批量升级，实现端到端的无感迁移；同时 x2openEuler 还支持对 Windows .NetFramework Web 场景进行业务无适配迁移，帮助客户把应用程序迅速迁移至 openEuler 上。

基于 openEuler 的迁移方案，包括成立迁移保障组织、迁移分析、方案设计、移植适配、迁移实施和测试上线六个阶段，如下图所示。

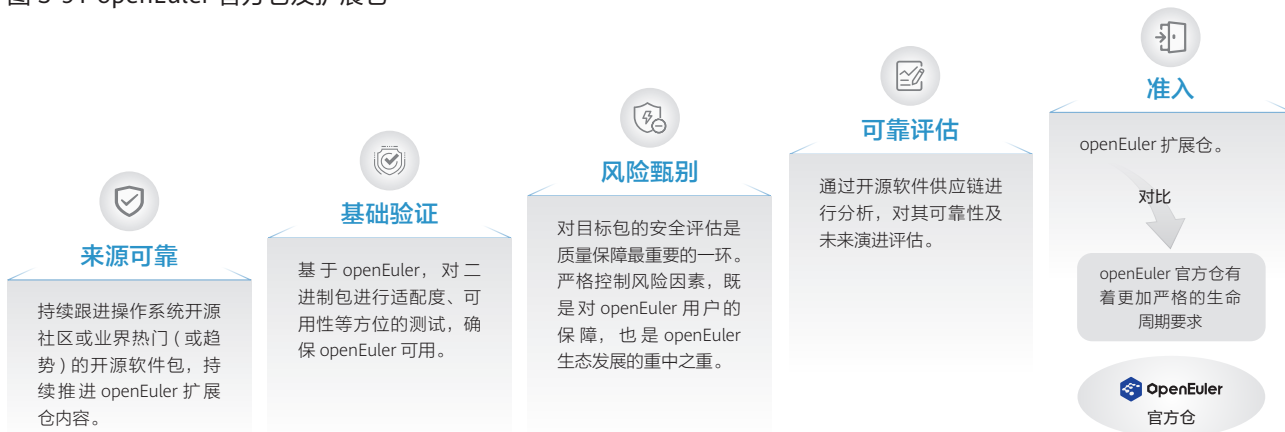
图 3-90 x2openEuler



### 3.3.3.3 openEuler 官方仓及扩展仓

openEuler 生态提供超过 3 万 + 源码包、百万级二进制软件包，可为 CentOS、Fedora 等系统向 openEuler 迁移的开发者、OSV、企业等用户提供一站式兼容性软件包、文件查询、下载、开源软件包的使用风险感知服务。

图 3-91 openEuler 官方仓及扩展仓



功能描述：

- » 面向创新场景开发者：提供快速的软件包引入平台，快速将上游项目形成 RPM 包。
- » 面向广大用户：提供 openEuler 已兼容的软件包查询，支持软件包、文件、命令查找。
- » 面向企业用户：提供开源软件风险一站式查询平台，感知软件活跃度、合规性、漏洞信息。
- » 面向企业用户：提供闭源软件分发渠道，支持企业软件直达用户。

# 04

鲲鹏原生开发

## 行业案例



# 4.1 金融

## 4.1.1

### 恒生电子

#### — 基于鲲鹏原生开发经纪场外业务系统，让证券数据处理更高效

恒生电子是聚焦金融行业，致力于为证券、期货、基金、信托、保险、银行、交易所、私募等机构提供整体 IT 解决方案和服务，包括财富管理、经纪业务等，连续 16 年入选 FinTech100 全球金融科技百强榜单，2023 年排名第 22 位，位列亚洲上榜企业第一。

#### 4.1.1.1 业务背景

数字化、智能化趋势下，证券用户的持续增长，交易数据处理量激增、交易响应速度要求达到微秒级，上一代系统在低延时、高吞吐、易扩展、安全方面已经无法满足需求。因此，证券企业也纷纷升级证券核心交易系统，以期为用户提供极速，稳定，低延迟的交易体验。

#### 4.1.1.2 鲲鹏原生方案

考虑到鲲鹏平台高吞吐、低时延、低能耗、高安全等优势，能够很好的满足恒生在证券行业业务的要求，此外还提供完善的原生开发工具链生态快速完成面向鲲鹏的迁移、开发、编译调试和性能优化；2024 年 3 月，恒生与鲲鹏达成了原生开发合作计划，首批选择新一代产品：经纪场外业务系统软件，融合鲲鹏原生开发技术实现从开发 - 构建 - 测试 - 发布 - 运行的全流程优化。

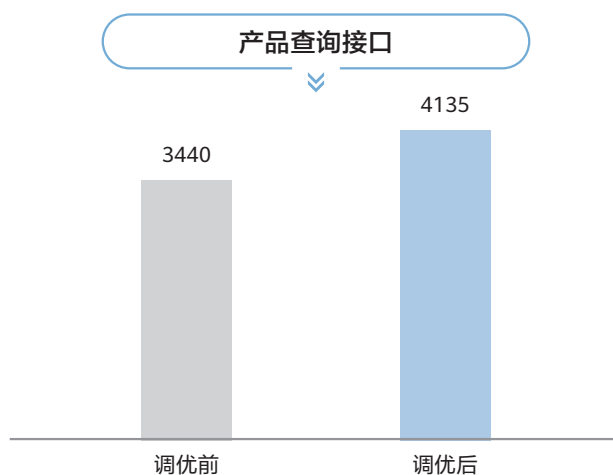
恒生将鲲鹏 DevKit 流水线技术栈融入自有的产品线管理、持续集成过程，形成新的构建集群。在恒生的流水线平台中，增加本次经纪场外业务系统的鲲鹏构建流程，基于鲲鹏服务器以及 openEuler 操作系统，搭建了持续集成节点。

其中，在构建流程中主要接入了鲲鹏 DevKit 门禁扫描、毕昇编译器、鲲鹏测试框架、性能分析工具等能力；在门禁扫描阶段可以直接使用 DevKit 迁移扫描工具快速的提示不兼容代码和软件包，确保代码在多样性的算力平台上兼容；在编译构建阶段，无感切换毕昇 JDK 优化编译性能；自动化测试阶段，通过 DevKit 提供的测试框架和样例来提高测试效率和软件质量，最后通过性能分析工具，进一步提升版本的性能。

### 4.1.1.3 业务价值

- » **原生高效率：**流水线改造完成后，开发人员可以只开发一套代码，有架构差异的代码使用宏定义隔离，每天快速在门禁阶段快速检查增量代码兼容性，并在不同平台上构建出不同的版本，效率更高。恒生基于改造后的流水线迭代经纪场外业务系统，整体开发效率提升 30% 以上
- » **原生高性能：**在编译构建阶段，流水线中能够无感知的将编译器替换为鲲鹏高性能编译器毕昇 JDK；同时在调优阶段，DevKit 提供的 Java 性能测试工具能够对软件运行性能进行采集和分析，通过分析智能合约运行容器 JVM 中内存占用和 GC 回收等瓶颈问题，结合虚拟化平台、数据库调优等优化措施，实现整体接口性能提升，其中产品查询接口的 TPS 性能提升 20%

图 4-1 优化前后性能对比图



经过上述一系列的优化，场外经纪业务系统的产品信息查询、产品定投流水、产品申购等业务接口的综合性能均得到明显提升，这些性能指标的优化，能够精准高效的处理证券业务信息的计算需求，让证券数据处理更加便捷和高效。

## 4.1.2

## 宇信科技

## — 基于鲲鹏原生开发金融软件，让金融信息处理更高效

北京宇信科技集团股份有限公司（简称：宇信科技）是国内金融科技的龙头企业，为超 1000 家金融机构提供包括咨询、产品、运营服务在内的一体化金融科技解决方案，在信贷操作系统、互联网信贷、监管合规、网络银行等细分市场占有率位居行业第一。

### 4.1.2.1 业务背景

随着 AI、大数据、5G 通信等新一代信息技术的日新月异和飞速发展，对金融行业数字化建设提出了更高挑战：包括大量个人信息、交易信息、系统状态等数据的传输和存储，对安全性和系统稳定性要求极高；大量线上业务常常产生高并发请求，处理能力如果达不到并发量要求，极易导致系统崩溃；各类监管报送模块迫切需要一套稳定、性能好、可扩展的统一监管报送平台。

### 4.1.2.2 鲲鹏原生方案

为了应对上述挑战，宇信科技选择携手鲲鹏共同推进金融科技自主创新实践。2024 年 2 月，宇信科技宣布启动鲲鹏原生开发计划，8 款核心软件将基于鲲鹏原生开发：

宇信科技 DevOps 平台团队基于现有云豹流水线深度集成鲲鹏 DevKit，包括迁移扫描、亲和分析、毕昇 JDK、调优工具和测试工具，通过设计和实施提交流水线、集成流水线和性能测试流水线，构建了一套全面的发布策略，确保代码变更快速、安全地部署到生产环境中；提交流水线负责快速验证每次代码提交，集成流水线则确保不同组件能够协同工作，而性能测试流水线则能够及早发现并解决性能瓶颈。

图 4-2 改造后的流水线



通过将鲲鹏 DevKit 原生开发与先进的 DevOps 实践相结合，不仅提高了开发效率，还增强了软件质量和可靠性，为鲲鹏平台上的应用开发和迁移设立了新的标准，为未来的创新和增长奠定了坚实的基础。

目前，统一监管报送平台 V5.0、分布式企业网银系统 V1.0、远程经营服务平台 V1.0 三款产品已完成鲲鹏原生开发，开发效率和产品性能均有显著提升。

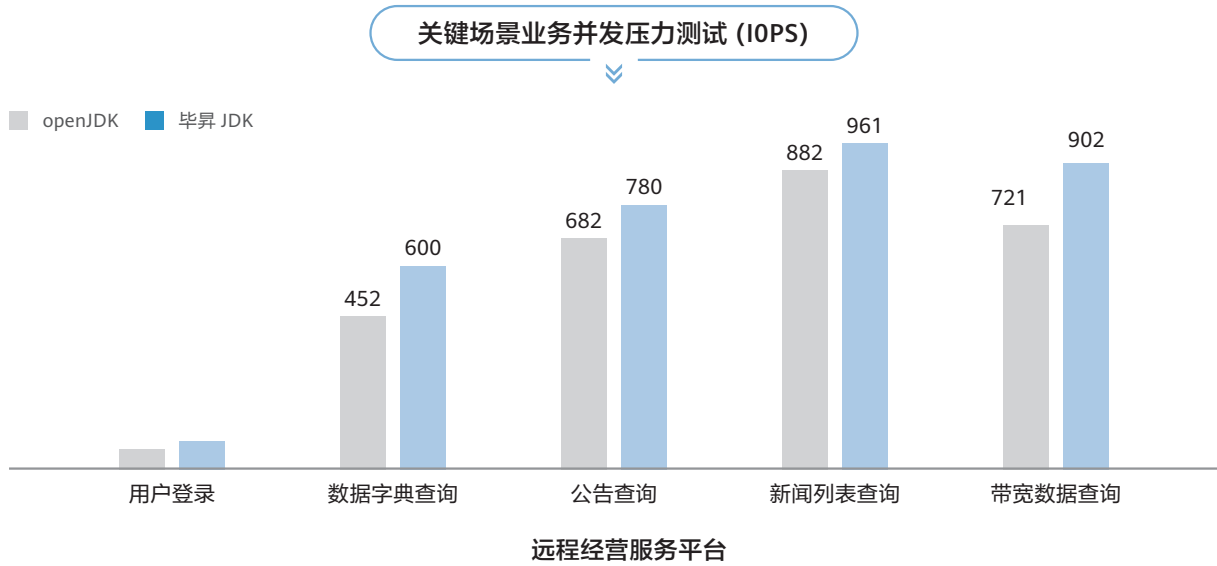
图 4-3 分布式企业网银系统 & 远程经营服务平台 鲲鹏原生架构图



### 4.1.2.3 业务价值

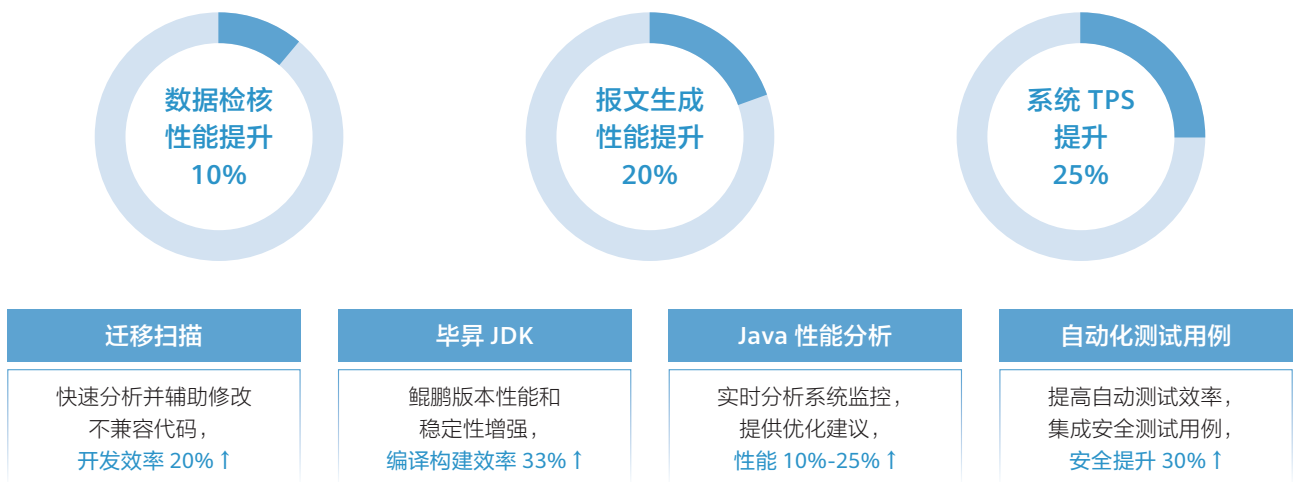
- » 原生高效率：改造之后的流水线使用率提升了 30%；同时在开发阶段通过鲲鹏 DevKit IDE 自动处理项目中的库依赖，包括版本冲突解决、依赖更新通知等，大大减轻了依赖库管理和维护上的负担；在测试阶段，鲲鹏 DevKit 会提供针对性能和兼容性的测试框架和用例，显著提升测试效率。最后，经对比分析，常规需求任务的开发效率提升了约 20%；
- » 原生高性能：通过无感知替换毕昇 JDK、以及通过 Java 性能测试工具分析和优化智能合约运行容器 JVM 中内存占用和 GC 回收等瓶颈问题，实现软件整体性能提升；
- » 分布式企业网银的会员服务切换毕昇 JDK 后应用启动耗时从 25s 缩短到 20s；
- » 远程经营服务平台选取用户登录、数据字典查询、公告查询、新闻列表查询、贷款数据查询等场景，分别使用毕昇 JDK 以及 OpenJDK 启动微服务集群开展压力测试，相同时间内，部分高并发场景的业务吞吐量提升 23%

图 4-4 切换鲲鹏毕昇 JDK 前后编译效率提升对比



» 统一监管报送平台的数据校核效率、报文生成性能、系统 TPS 均有 10~25% 的提升，这些指标的提升让银行的监管报送更加高效；

图 4-5 统一监管报送平台基于鲲鹏 DevKit 的优化效果



目前，宇信科技的分布式企业网银系统和远程经营服务平台目前已在多个银行落地实施，通过高效的信息整合和处理，显著提升了企业信息处理和业务运营效率，让银行经营更高效，用户服务更满意。

# 4.2

## 电信

### 4.2.1

#### 浩瀚深度

##### — 鲲鹏原生赋能 DPI 采集系统，网络数据可视化处理更加高效

北京浩瀚深度信息技术股份有限公司（以下简称“浩瀚深度”）深耕在互联网流量和数据智能化领域，为运营商及政府提供采集存储分析应用的垂直一体化服务，是网络和数据智能化的领先企业。其高性能、高精度、高可靠的整体解决方案助力互联网行业实现了网络可视、资源优化、智能管控、安全防护及数据提升，推动了互联网大数据普及与应用。

#### 4.2.1.1 业务背景

数字化时代，互联网流量与数据智能化已跃升为行业发展的核心驱动力，高效管理互联网流量、确保数据安全无虞，并成功实现数据的智能化应用，已成为当前互联网领域亟需攻克的重大课题。浩瀚深度的核心产品浩智睿— DPI 采集系统，采用多核分布式架构，提供对流量数据进行协议解析、业务识别、流控和封堵等功能，应对日益严重的安全挑战，实现流量及数据的智能化应用。

随着物联网、移动互联网、5G 等技术的快速发展，用户对于网络和数据智能化业务的并发量提出了新的诉求，特别在运营场景下可产生千万量级的高并发请求，会对处理能力低的系统造成冲击。

#### 4.2.1.2 鲲鹏原生方案

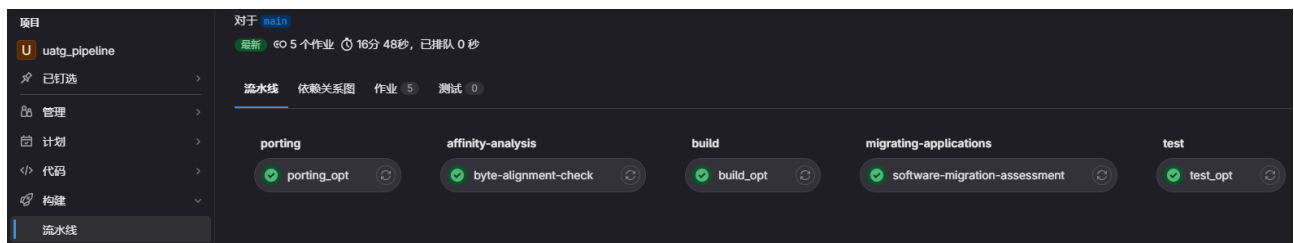
为了进一步优化系统性能、敏捷迭代版本以应对瞬息万变的互联网环境，浩瀚深度携手鲲鹏共同推进网络和数据智能化业务应用的鲲鹏原生创新转型建设：基于鲲鹏硬件、openEuler OS、鲲鹏 DevKit、鲲鹏 BoostKit，实现 1 套代码 +1 条流水线 -> 多平台版本，持续且首发性能领先的鲲鹏商用版本，目前浩智睿— DPI 采集系统 V3.7.0 版本已经完鲲鹏原生优化改造。

图 4-6 浩智睿一 DPI 采集系统架构图



浩瀚深度将鲲鹏 DevKit 流水线技术栈融入自有的产品线管理、持续集成过程，在流水线中直接调用迁移扫描和亲和检查门禁、毕昇编译器、鲲鹏测试框架、性能分析插件、版本漏洞扫描插件等 DevKit 工具能力，从而提升版本的性能、测试效率和质量；并基于鲲鹏服务器以及 openEuler 操作系统，搭建了持续集成节点，鲲鹏软硬协同优化确保软件高性能运行。

图 4-7 浩瀚深度改造后的流水线信息

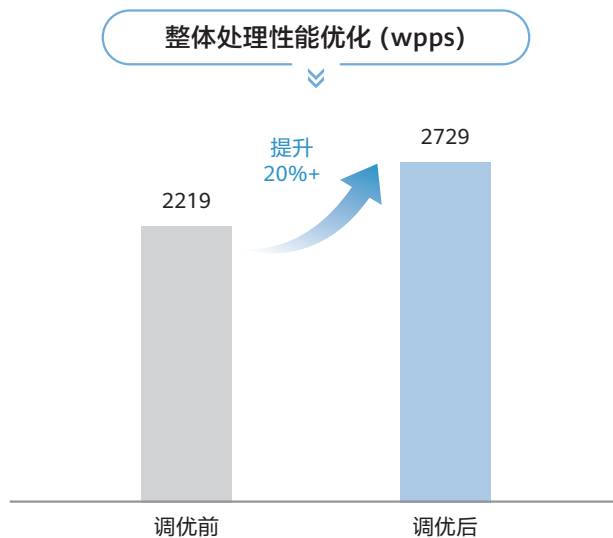


- » 在代码开发时使用鲲鹏 DevKit IDE 插件提供的场景化 SDK 快速完成应用的初始化、开发及部署，自动提示和补充鲲鹏的高性能函数，加快了代码开发效率；
- » 门禁检查阶段，对增量代码的鲲鹏兼容性和鲲鹏架构亲和性进行自动检查和快速优化，有架构差异的代码使用宏定义隔离，加快代码适配速度；
- » 编译构建阶段，将开源 GCC 切换为鲲鹏高性能编译器 GCC for openEuler，以流量分发模块为例，单核 PPS 从 218 万提升到 223 万
- » 测试阶段，在流水线自动调用鲲鹏测试框架及测试用例，快速完成性能与兼容性的测试，测试效率提升 10%；
- » 调优阶段，DevKit 提供的性能测试工具对软件运行性能进行采集和分析，对程序热点瓶颈进行了分析和优化；

### 4.2.1.3 业务价值

- » 原生高效率：经对比分析，常规需求任务的开发平均用时从 3 人日缩短至 2 人日，效率提升 30+%；
- » 原生高性能：鲲鹏原生改造后，浩瀚深度的浩智睿一 DPI 采集系统在业务识别、输出流量日志、文件还原、封堵以及镜像功能等方面实现了显著的性能提升。其中，智能化采集业务的综合数据处理能力提升 20%+，这些性能指标的优化，让网络数据可视化处理更加高效便捷。

图 4-8 智能化采集业务基于 DevKit 优化前后性能对比图



目前，该解决方案已广泛服务于运营商、金融及政府部门等行业，不仅提供强大的数据支持能力，还能面向用户的快速需求变化进行迭代，提高应用的效率和质量，降低运维成本和复杂度，加速行业数字化转型。

## 4.2.2

### 恒安嘉新

#### — 基于鲲鹏原生开发安全采集分析平台，保障企业信息系统安全稳定运行

恒安嘉新(北京)科技股份有限公司(简称:恒安嘉新)专注于通信网和互联网数据治理和安全防护,为行业用户提供“云网边端业数智安”综合解决方案,致力于成为智慧城市数智基建搭建者和全连接服务运营者。

#### 4.2.2.1 业务背景

恒安嘉新的核心产品之一安全采集分析平台能够实时捕获恶意流量、集中分析研判,对公共互联网中移动恶意程序进行监测发现、样本还原、研判分析、处置防护等处理,解决移动核心网络上多种类别的恶意程序传播问题,消除网络安全隐患,降低网络安全风险。

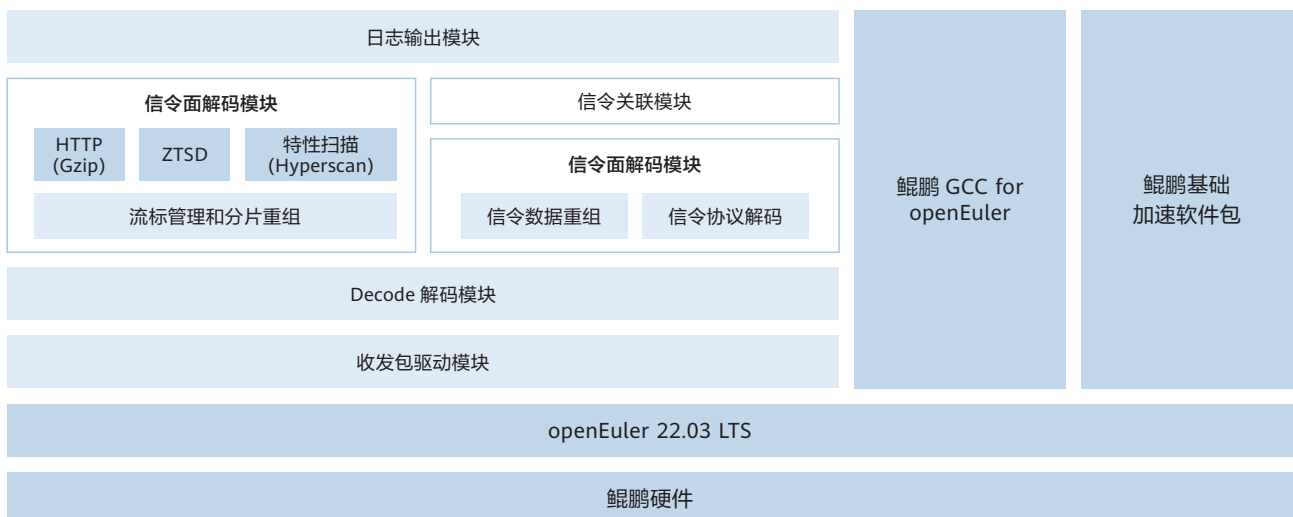
随着以“大流量”、“大连接”为特征的新一代通信技术的不断发展和大规模应用,用户对于流量监测产品提出了更高的诉求:

- » **高吞吐:** 应用的多样化导致网络数据激增,为了满足流量实时采集、传输、存储、分析等需求,迫切需要提升海量数据在线计算和分析的能力;
- » **高实时:** 未来的流量单位都是 TBPS,为了提高响应速度,并发连接数不少于 5000 万,新建连接不低于每秒 160 万。

#### 4.2.2.2 鲲鹏原生方案

为了实现上述目标,恒安嘉新选择基于鲲鹏软硬件平台打造安全采集分析平台一体机解决方案,结合鲲鹏天然的多核高并发、高安全等优势进一步增强稳定性和可用性,同时,通过结合鲲鹏开发套件 DevKit,提升开发效率,优化流量分析处理专项指标。

图 4-9 恒安嘉新安全采集分析平台方案



恒安嘉新在自有的 Gitlab 流水线中充分融入鲲鹏 DevKit 迁移扫描、编译、调试、性能分析等能力，流水线改造完成后，开发人员可以只开发一套代码，有架构差异的代码使用宏定义隔离，每天快速在门禁阶段检查代码兼容性，并在不同平台上构建出不同的版本，效率更高。

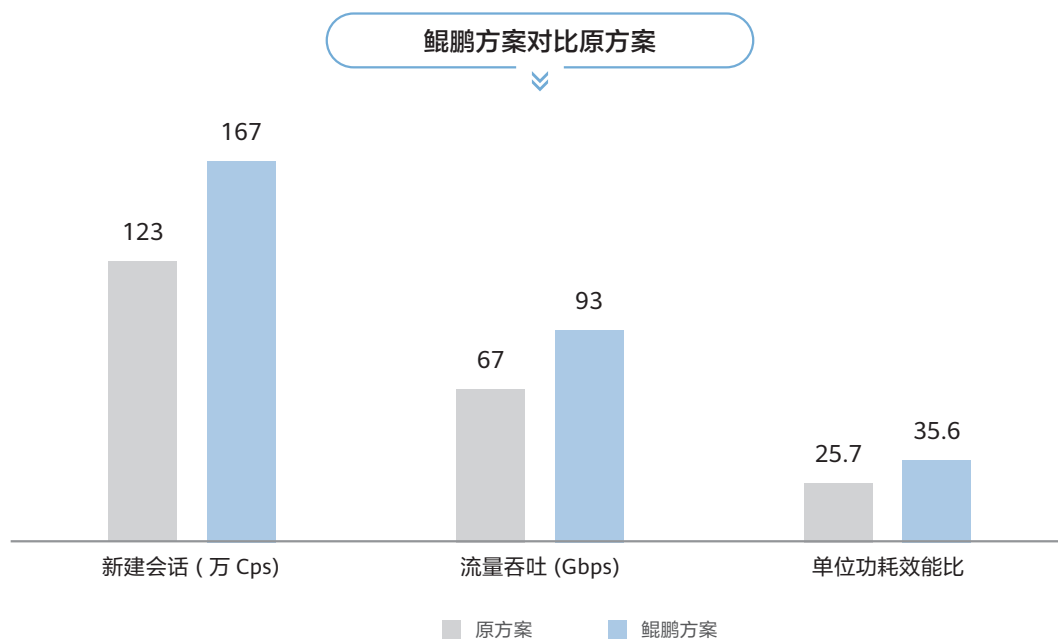
在编译过程中，由于 Linux 服务器一般不提供 GUI 界面，因此编译和调试只能基于命令行，效率低、门槛高。鲲鹏 DevKit 可实现远程图形化的调试，一键完成本地代码到远程服务器的代码同步以及编译构建过程，降低了技术人员上手门槛。

调优过程中，开发人员还使用了鲲鹏 DevKit 的调优助手和系统性能分析工具进行了相关调优：通过系统配置全景功能配合 PCIe 拓扑图功能，帮助技术人员快速将硬件拓扑、BIOS、系统层面等调至最优状态。同时，通过处理器微架构分析帮助技术人员快速锁定性能瓶颈点，定点突破。

### 💎 4.2.2.3 业务价值

- » 原生高效率：流水线改造后，新版本开发效率提升 20%。
- » 原生高性能：经过鲲鹏 DevKit 提调优之后，安全采集分析平台的系统配置功能整体性能提升了 39%

图 4-10 系统配置全景功能基于鲲鹏 DevKit 优化前后对比



当前，恒安嘉新安全采集分析平台一体机解决方案已广泛应用于移动互联网、城域网、数据中心、交换中心、直联点等核心网络节点。在运营商恶意程序系统、新型交换中心等多个项目中得到充分应用，帮助用户在大流量、高并发场景下完成网络流量的安全监测，有效降低运维成本和复杂度，加速实现数字化转型。

# 4.3 政府

## 4.3.1

### 超图软件

#### — 基于鲲鹏原生开发地理信息系统，让自然资源评估更高效

北京超图软件股份有限公司（以下简称“超图软件”）是自然资源领域领先的解决方案供应商，聚焦地理信息软件（Geographic Information Software，广义 GIS）和空间智能（Geospatial Intelligence，GI）领域的基础软件与应用，其自主研发的 SuperMap 软件是亚洲第一大、全球第二大 GIS 软件，是数字中国、数字政府、企业数字化、数字孪生、元宇宙、智慧城市的重要技术底座。

#### 📄 4.3.1.1 业务背景

近年来全球各地自然灾害、地质灾害频发，GIS 能够快速处理和分析海量空间、属性数据。在灾害应急救援过程中，全面深度分析各类地理信息、应急资源、灾害现场数据，为灾后救援提供有力支持。为确保科学有效防灾减灾，SuperMap GIS 借助多源数据融合手段搭建应急管理数字底座，助力政府部门高效应急防范，在风险评估与情景模拟、突发事件快速响应、灾后重建规划等领域发挥积极作用。

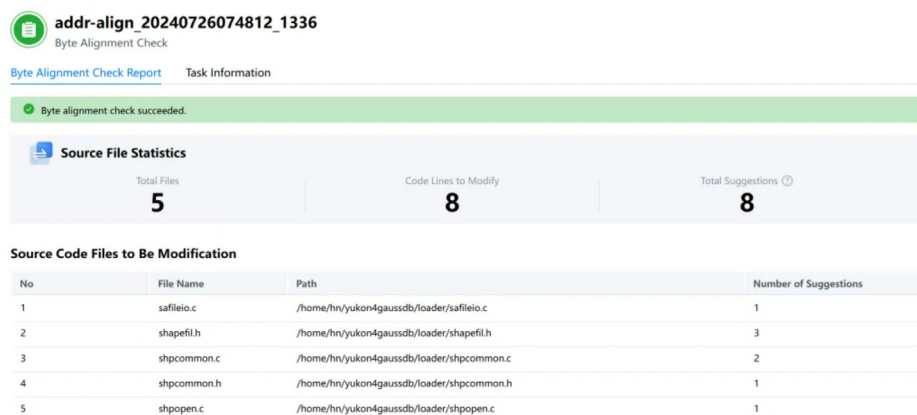
#### 🏠 4.3.1.2 鲲鹏原生方案

为了全面推进地理信息产业数字化的自主创新进程，打造高性能、可持续的解决方案，超图软件携手鲲鹏针对 SuperMap GIS 系列内五款核心产品进行了原生开发改造和优化：

- » 超图服务器 GIS 软件平台 SuperMap iServer；
- » 超图 GIS 门户软件平台 SuperMap iPortal；
- » 超图 GIS 边缘软件平台 SuperMap iEdge；
- » 超图组件式地理信息开发平台（Spark 版）SuperMap iObjects for Spark；
- » 禹贡空间数据库管理系统 SuperMap YukonDB。

超图软件在自有 CI/CD 流水线中接入了鲲鹏 DevKit 代码兼容性检查门禁，编译器从 openJDK 替换为毕昇 JDK，此外还接入鲲鹏性能分析插件、鲲鹏测试框架等，改造后的流水线可以持续发布鲲鹏版本，不需要手动验证，可以在 CI 上自动运行检查、发布程序安装包，将 SuperMap GIS 在稳定性、兼容性等方面提升到了新的高度，同时还能更加便捷的对应用程序进行性能优化，更好地满足用户在复杂场景下的应用需求。

图 4-11 迁移扫描门禁检查报告

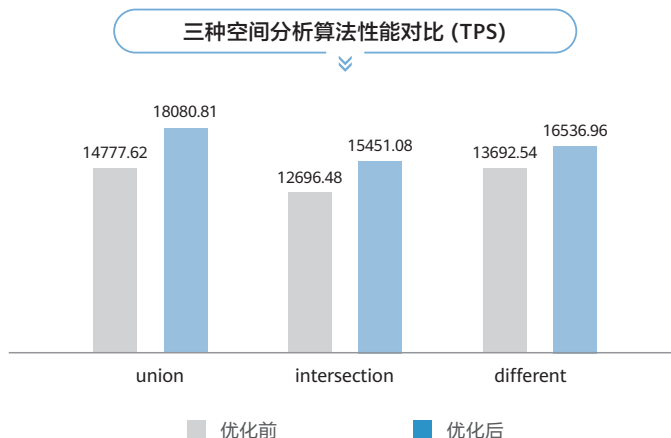


此外,在调优过程中开发人员还充分运用了鲲鹏 DevKit 性能分析工具对热点函数、软硬件配置参数等进行分析优化,运用 BoostKit 加速算法,包括 KAE 加速、NUMA 调度优化、加速指令优化等对数据库进行调优。

### 4.3.1.3 业务价值

- » 原生高效率: 基于鲲鹏原生开发改造后的流水线进行 SuperMap GIS 新版本迭代,整体开发效率提升 25%。
- » 原生高性能: 以禹贡空间数据库管理系统 YukonDB 为例,经过 GCC for openEuler 编译优化,整体性能提升 30%+;其中,通过 DevKit 对硬件层面的 BIOS 设置优化,在 OLAP 场景 32 线程并发下,常用 TOP3 GIS 空间分析功能(合并 union、交集 intersection、差集 different) TPS 性能均有 20% 以上的提升。

图 4-12 YukonDB 常用 TOP3 GIS 空间分析功能性能优化对比



基于鲲鹏原生开发的 SuperMap GIS 能够更好的应对海量数据和复杂空间数据处理和分析的挑战,尤其是在面向资源调查、确权登记、规划等应用领域中,用户可以更快地进行数据处理、分析和决策,降低投入时间,在实时或近实时分析场景上,软件效率的提升可以确保用户及时获取结果,用户体验更佳。

目前 SuperMap GIS V11 (鲲鹏原生版本) 已经在水利行业商用落地,实现水利空间业务数据管理、水利一张图系统高性能、稳定运行;禹贡空间数据库(鲲鹏原生版本)也已在自然资源行业商用落地,实现亿级数据秒级检索和十亿级图斑的实时统计分析。

## 4.3.2

# 深圳防灾减灾技术研究院

## — 鲲鹏原生助力地震观测数据高效处理

地震监测预警事业与国家经济发展、社会稳定和国民的生命财产安全息息相关，自国家地震烈度速报与预警工程建设项目开展以来，已建设一万五千多个地震监测台站，可在震后 5-10 秒产出地震预警信息，而实时汇集传输和存储管理这些台站全天候产生的海量监测数据成为地震台网的一项重要工作。

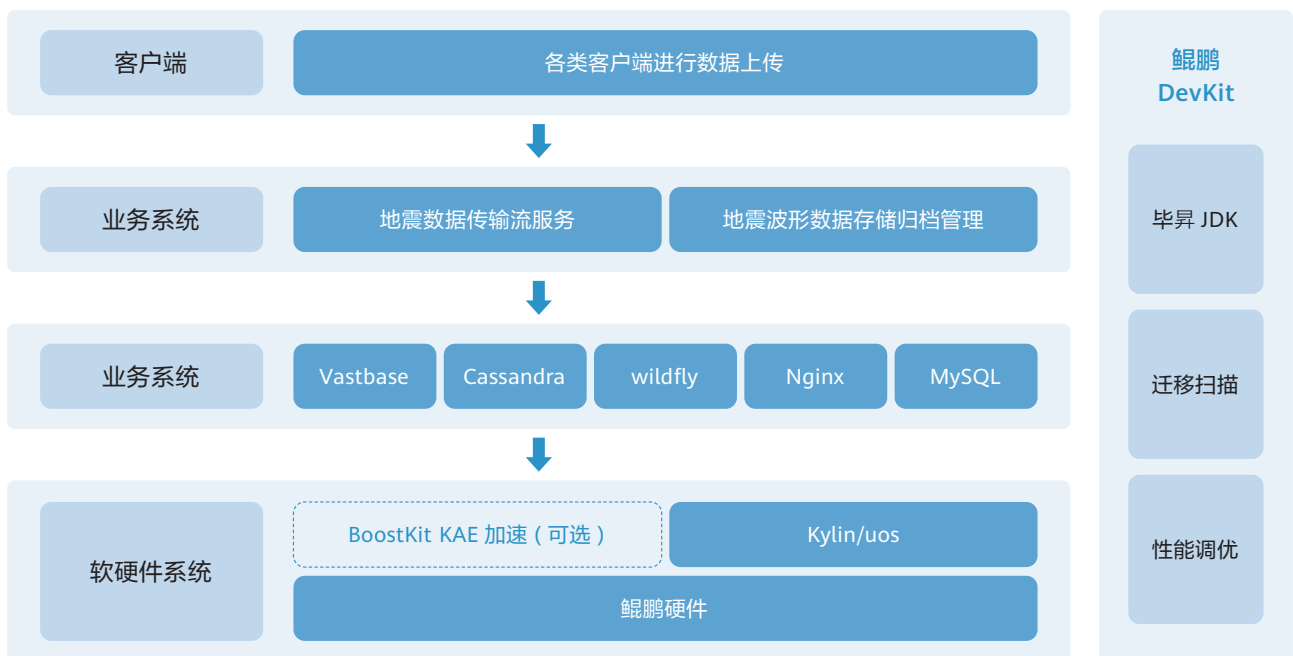
### 4.3.2.1 业务背景

深圳防灾减灾技术研究院聚焦预警工程定制软件研发，致力于达到实时传输汇集、存储处理近两万个地震台站的海量实时数据，以及在震后几秒内测定地震参数并向社会发布地震预警信息的目标。

### 4.3.2.2 鲲鹏原生方案

为了更好的支撑地震传输流服务和地震波形数据管理业务，深圳防灾减灾技术研究院携手鲲鹏原生开发地震数据传输流服务软件 SeisWave-SSS 和地震波形数据存储归档管理软件 SeisWave-AWS：在现有 CICD 流程中增加本次地震业务软件的鲲鹏构建流程，实现从开发 - 构建 - 测试 - 发布 - 运行的全流程优化。

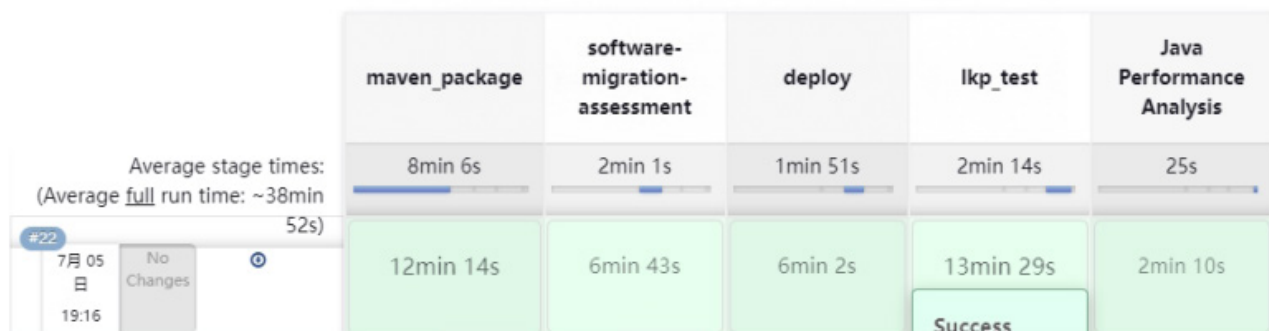
图 4-13 SeisWave-SSS 和 SeisWave-AWS 鲲鹏原生架构图



开发人员基于鲲鹏服务器以及麒麟操作系统（openEuler 系），搭建了持续集成节点；在构建流程中融入了鲲鹏 DevKit 流水线技术栈，包括流水线门禁、毕昇编译器、鲲鹏测试框架、性能分析插件、版本漏洞扫描插件等能力，确保构建产物与鲲鹏架构的适配性，全面优化在鲲鹏架构上的开发及运行效率，还能进一步提升版本的性能、测试效率和质量。

图 4-14 改造后的流水线

### 阶段视图



流水线优化改造后，开发人员只需开发一套代码，有建构差异的代码会使用宏进行定义隔离，在门禁阶段可快速检查代码的兼容性和亲和性，并构建不同平台版本，提升开发质量的同时保证代码质量。

### 💡 4.3.2.3 业务价值

- » 原生高效率：基于改造后的 Jenkins 流水线，开发人员只需提交开发好的代码，流水线会自动进行门禁检查、编译构建、部署测试等工作，无需人工干预，大大节省了人力成本。经对比分析，开发效率提升了 50% 以上。
- » 原生高性能：在编译构建阶段，流水线中会无感切换为毕昇 JDK，该编译器针对鲲鹏架构进行了多维度优化提升，基于 AppCDS 特性、快速序列化、NUMA-Aware 特性等特性，将进一步提升应用的运行效率。经测试，以传输时延持平为前提条件，鲲鹏服务器使用的 CPU 资源更少，理论上能够提升 10% 的吞吐率。

目前，SeisWave-SSS 和 SeisWave-AWS 已经在我国地震预警台网广泛应用，支撑我国地震预警网海量实时观测数据的汇集传输和存储管理，实现了基于鲲鹏 +openEuler 系操作系统的自主创新改造和长期稳定运行，保障了我国地震观测数据的安全。除了应用于地震预警监测之外，软件还可以用于流动地震科学观测、水库地震安全监测、地震应急监测、石油勘探地球物理监测、建筑物振动监测、铁路振动监测、桥梁振动监测、核爆及爆破监测等领域，有着广泛的商业应用价值。

# 4.4 电力

## 4.4.1

### 广州海颐

#### — 基于鲲鹏原生开发高性能、高可靠电力系统软件，加速电网数字化转型

广州海颐软件有限公司（简称“广州海颐”）是一家面向全国提供企业及政府信息化解决方案与服务的公司，通过打造全链条电力能源数字化解决方案，帮助行业实现业务流程化、管理精益化、运营可视化，助力国家能源转型与数字经济发展。

#### 4.4.1.1 业务背景

5G、大数据、工业互联网等技术与能源电力产业的加速融合，产生了大量数据要素，如何利用好这些数据为政府、能源大企业、中小企业、用户等打造信息互通、价值共享的平台生态，并通过数字化创新手段来提升运营效率、降低能耗排放、优化资源配置、实现与绿色化市场的融合发展，成为能源电力企业数字化转型进程中的迫切需求。

#### 4.4.1.2 鲲鹏原生方案

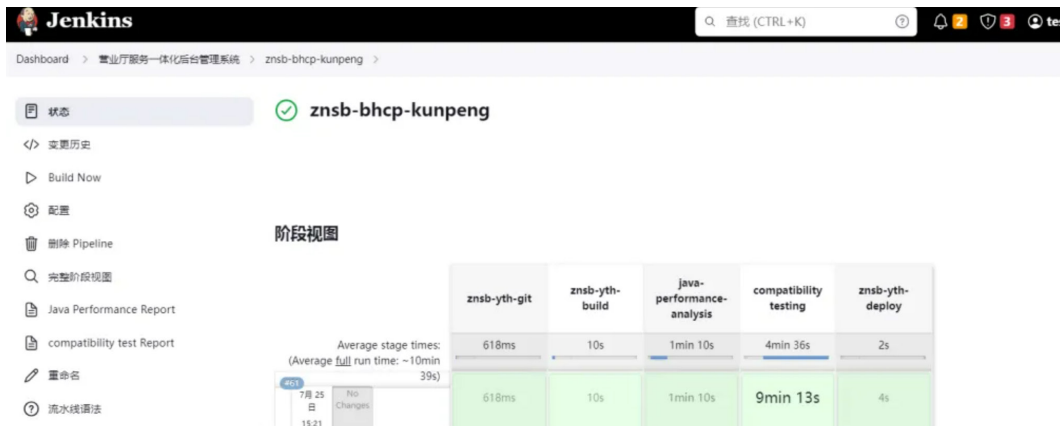
为了打造高性能、高可靠电力系统软件，加速电网数字化转型，广州海颐首选 5 款核心产品基于鲲鹏原生开发：购电全链路监控 V1.0.2、绿色用能应用 V1.0.1、供电服务进社区 V1.2.1、营销用电地址管理 V1.2.2。

这些软件聚焦完善、整合、管理政府、企业和社区范围内的能源电力数据，更加精准地了解并满足客户需求，助力企业新业务精准推广，实现电力设备管理可视化、用户数据处理绿色化、政企服务流程规范化，激发行业应用与社会治理协同发展。



在原生开发过程中，海颐在自有 Jenkins 开发流水线中融入鲲鹏 DevKit 技术栈，包括源码扫描门禁、毕昇 JDK、Java 性能分析、兼容性测试等能力，确保构建产物与鲲鹏架构的兼容适配性，提升版本的性能、测试效率和质量。同时，基于鲲鹏硬件以及 openEuler 操作系统，搭建了持续集成节点，全面优化软件在鲲鹏架构上的运行效率。

图 4-15 改造后的广州海颐开发流水线

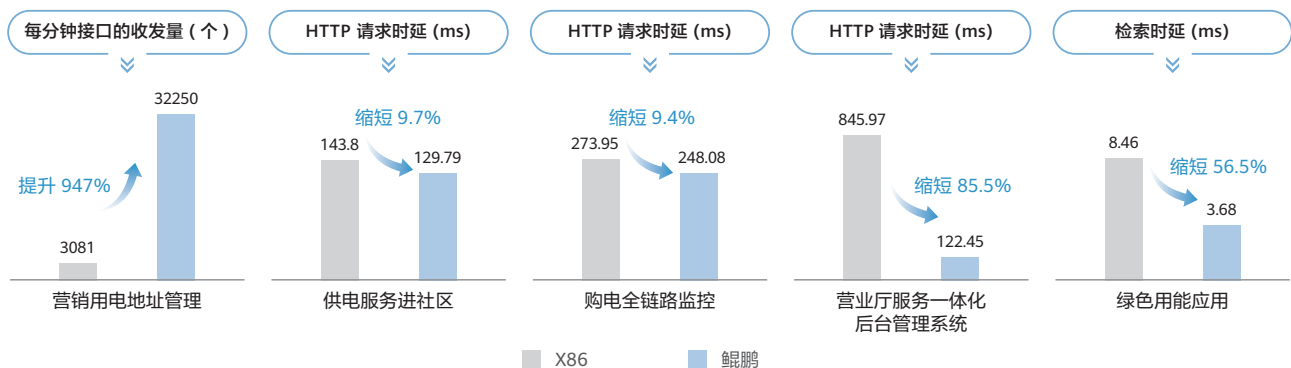


### 4.4.1.3 业务价值

5 款软件基于改造后的流水线迭代新版本，兼容性、性能、开发效率等都得到优化提升：

- » 原生高效率：开发流水线进行软件开发和优化，实现一套代码多平台产品发布能力，开发调试效率 25% 以上的提升；
- » 原生高性能：通过 DevKit Java 性能分析功能对软件运行性能进行采集和分析，以及操作系统参数调优等手段，应用在压力负载下，软件的接口容量和时延均有显著提升，其中接口时延平均缩短 40.2%；

图 4-16 5 款软件原生开发改造后性能对比



这些指标的提升，让广州海颐能更加快捷、精准的制定电力营销策略，进一步优化电力资源配置，提升用电报装效率、居民响应速度、电力营业厅运营管理效率等，让企业服务流程更完善，居民供电服务更满意。

## 4.4.2

### 国能信控

#### 一 基于鲲鹏原生开发新能源计算平台，为电力系统提供高效算力支撑

国能信控技术股份有限公司（以下简称“国能信控”）深耕于新能源工控安全智能化建设领域，业务覆盖新能源工控及配套系统、智慧能源综合解决方案两大板块。新能源计算平台作为国能信控的核心产品，承担区域新能源指标计算业务，并为生产智能监控系统和报表系统提供数据支撑，实现新能源发电产业的数字化转型和智能化升级。

#### 4.4.2.1 业务背景

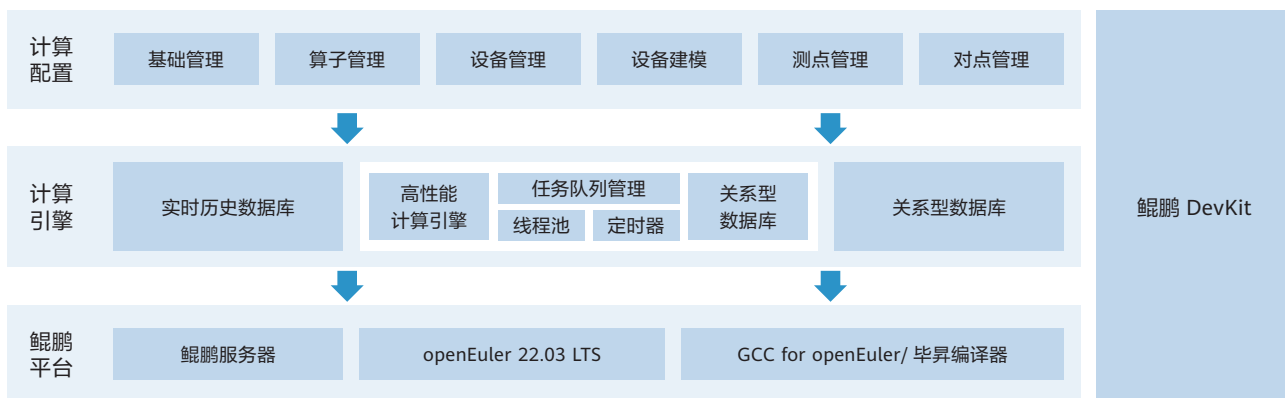
随着能源行业向数智化、低碳化发展，其产业结构、技术体系架构和行业标准规范等发生深刻变革，加快构建清洁低碳、安全充裕、经济高效、供需协同、灵活智能和自主创新的新型电力系统成为能源行业转型的当务之急。在物联网、5G 通信、人工智能等新技术的大规模应用的背景下，新能源实时计算平台正面临着多重挑战：

- » 海量数据处理：在线处理的实时数据正呈现出海量化、高频化、异构化的特点，计算平台的大数据处理能力，直接影响系统的业务容量和业务支撑范围；
- » 高实时性计算：部分新能源业务场景下，要求数据计算任务的执行周期达到毫秒级，这项能力对于实现系统状态实时监控以及高频低延时下控功能起到关键作用；
- » 稳定运行：新能源发电具有间歇性、不稳定性等特点，部署场景也具有多样性，给电力系统的安全稳定运行带来更高要求。

#### 4.4.2.2 鲲鹏原生方案

为应对上述挑战，打造安全高效的指标计算业务核心产品，国能信控携手鲲鹏开启新能源计算平台的原生开发创新实践。对自有流水线进行了优化改造：在开发 - 构建 - 测试 - 发布 - 运行的全流程中接入鲲鹏 DevKit 流水线门禁系统、毕昇编译器、系统调优、测试框架等工具等能力，提升代码质量和开发效率；并以鲲鹏服务器以及 openEuler 操作系统为基础，搭建了持续集成的环境，确保软件高性能运行。

图 4-17 新能源计算平台基于鲲鹏的原生开发框架



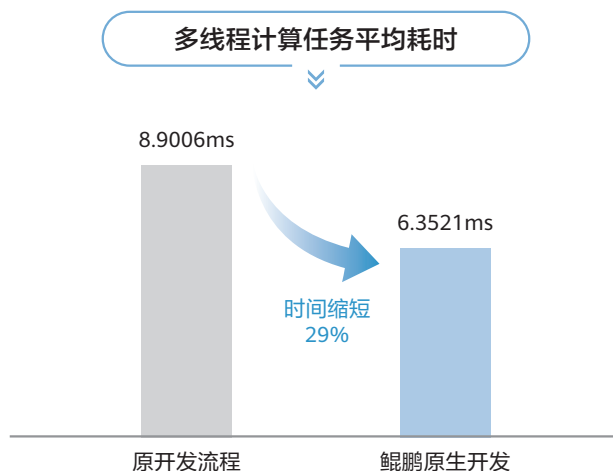
基于改造后的流水线，开发人员对新能源计算平台 V2.0 进行新版本迭代：

- » 代码开发阶段，使用鲲鹏 DevKit IDE 插件提供的场景化 SDK 提升了编程的自动化、智能化水平，提高了研发效率；
- » 门禁检查阶段，流水线中的鲲鹏 DevKit 门禁检查系统对存量代码进行全面扫描，并根据提示信息进行修改和优化，有架构差异的代码使用宏定义隔离，不仅减轻了开发人员的工作负担，也让代码更加规范，执行更加高效；
- » 编译阶段，鲲鹏原生开发平台支持从开源 GCC 到毕昇编译器的平滑切换，对 C++ 代码进行编译优化；
- » 调优阶段，DevKit 提供的系统性能分析工具对软件运行性能进行采集和分析，通过进线程分析，热点函数分析，微架构分析，IO 分析，锁与等待分析，内存诊断等瓶颈问题，实现整体性能的提升。

### 💎 4.4.2.3 业务价值

- » 原生高效率：新能源计算平台版本 V2.0 的开发效率对比原平台提升了 10% 以上。
- » 原生高性能：以高效计算引擎为例：使用原本本进行某项多线程计算任务时，平均耗时为 8.9006ms，而采用基于鲲鹏原生开发平台开发且使用毕昇编译器生成的版本，执行相同任务的平均耗时为 6.3521ms，平均耗时缩短近 30%。

图 4-18 高效计算引擎多线程计算任务平均耗时对比



目前，新能源计算平台版本 V2.0 已获得鲲鹏原生开发认证，并在多个新能源区域集控项目和新能源智慧场站项目中部署并使用，为生产监控平台、报表系统等提供算力支撑。

国能信控积极响应国家政策战略导向及能源集团自主创新的号召，后续更会逐步安排其他相关产品在鲲鹏原生开发平台上进行测试与应用，将鲲鹏平台的优秀实践推广到更大的业务范围和更多的应用场景中，全面推进新能源计算平台的自主创新建设。

# 4.5 医疗

## 4.5.1

### 卫宁健康

#### — 基于鲲鹏原生开发新一代医疗信息系统 WiNEX，让医疗服务更便利

卫宁健康科技集团股份有限公司（简称：卫宁健康）专注于医疗健康和卫生领域的整体数字化解决方案与服务，业务覆盖临床诊疗、医院管理、健康服务、医学科研、公共卫生、互联网医疗等，是国内领先的医疗健康信息化企业之一。

#### 4.5.1.1 业务背景

医疗健康是公共服务领域的重要行业，关乎国计民生。随着医疗卫生信息化的快速发展，医院管理、医疗服务、诊疗方式与过程正发生着深刻变革，加快实现智慧医疗、医疗信息安全建设刻不容缓。

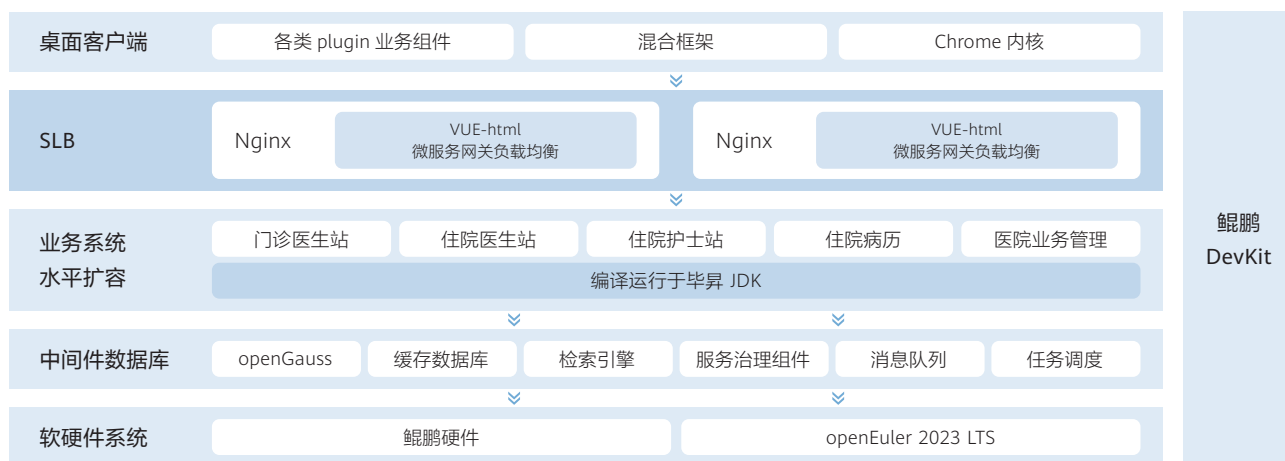
#### 4.5.1.2 鲲鹏原生方案

卫宁健康携手鲲鹏共同推进医疗核心应用的自主创新转型建设，首批选择新一代产品 WiNEX 中五个医疗核心系统：医院业务管理、门诊医生站、住院医生站、护士工作站、住院病历管理软件，基于鲲鹏原生应用开发进行技术改造。

在卫宁健康自有 DevOps 平台中，增加本次医疗核心应用的鲲鹏构建流程，基于鲲鹏服务器以及 openEuler 操作系统，搭建了持续集成节点。在构建流程中接入鲲鹏 Devkit 流水线门禁，通过该门禁进行应用亲和性扫描以及代码迁移扫描，确保构建产物与鲲鹏架构的适配性，全面优化在鲲鹏架构上的开发及运行效率。



图 4-19 WiNEX 基于鲲鹏原生应用开发框架

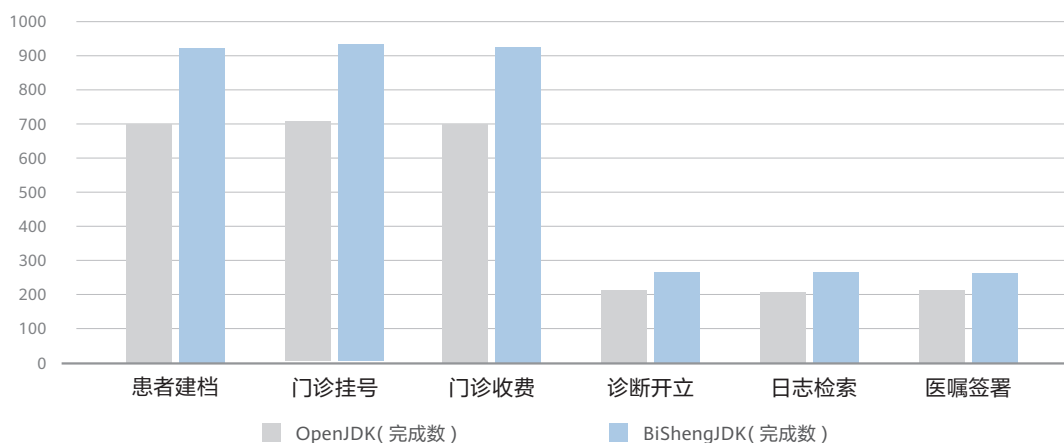


应用开发阶段，使用鲲鹏 DevKit IDE 提供的场景化 SDK 和启发式编程，提升了研发人员在初始化、开发及部署方面快速完成需求的能力；在构建原生应用时，自动将任务调度至鲲鹏服务器节点，使用毕昇 JDK 进行程序构建；编译完成后，调用鲲鹏 DevKit 流水线门禁检查组件，进行鲲鹏亲和性检查分析；同时，集成鲲鹏的兼容测试工具，对构建结果自动进行启停测试、安全检测、资源使用检测等，确保对鲲鹏环境的兼容性。

### 4.5.1.3 业务价值

- » **原生高效率：**经对比分析，基于改造后的流水线常规需求任务的开发平均用时从 3 人日缩短至 2 人日，效率提升约 30%。
- » **原生高性能：**在编译构建时将 OpenJDK 无感替换为性能更优的毕昇 JDK，选取患者建档、门诊挂号、诊断开立、西药、草药、检验、检查等医嘱开立及签署等场景，针对两款编译器开展压力测试，相同时间内，部分高并发场景的业务吞吐量提升 33%。

图 4-20 固定时间内不同场景的压力测试对比



卫宁健康将持续深化医疗核心业务鲲鹏原生应用开发，通过建立更稳定、更安全的医疗信息技术软硬件基础，推动医疗全栈生态体系建设，从而真正实现医院核心业务的性能提升，让其不仅“能用”，更要“好用”。

## 4.5.2

## 东华软件

## — 构建鲲鹏原生医疗经营管理系统，让医院运作更有序

东华软件股份公司（简称：东华软件）作为国家级高新技术软件企业，以“科技呵护健康、孪生助力成功”为使命，专注于构建城市智慧医疗操作系统（CMOS）与医院综合业务操作系统（HOS），为大健康产业提供完整和闭环、线上和线下相结合的软硬件解决方案。

### 4.5.2.1 业务背景

随着医院管理、规模及业务的快速发展与变革，业务与管理复杂度与日俱增，包括医院绩效管理、跨机构互联互通，以及临床业务全流程闭环管理等需求，传统医疗 IT 系统的产品化建设及更新迭代模式显得力不从心，具体表现为：新功能上线周期长、迭代更新速度慢；开发维护成本变高、部署效率降低；技术兼容性差、生态不完善、安全风险无法把控等问题。

### 4.5.2.2 鲲鹏原生方案

东华软件联合鲲鹏共同推进医疗核心应用的自主创新建设，首选医院综合业务操作系统（HOS）开展原生开发技术改造。医院综合业务操作系统（HOS）作为整个医院信息化系统的入口和展现平台，精准映射医院全域业务流程，HOS 门户系统支持按岗位定制工作页面，确保核心模块统一而次要模块个性化，推动医院管理向智能化、高效化迈进。

东华软件在原有 DevOps 平台中融合鲲鹏原生开发技术，开发人员将代码提交流水线后，可直接调用鲲鹏 DevKit 门禁检查、编译、调优、测试、版本安全检查等能力。其中，在编译构建阶段，将编译器替换为毕昇 JDK；调优过程中，使用了鲲鹏 DevKit 提供的系统性能分析和 Java 性能分析工具，优化服务器系统参数、JVM 内存和垃圾回收等相关配置；版本发布时，使用增量 / 版本级编码扫描能力，对编码规范、漏洞进行扫描，进一步保障版本的质量。



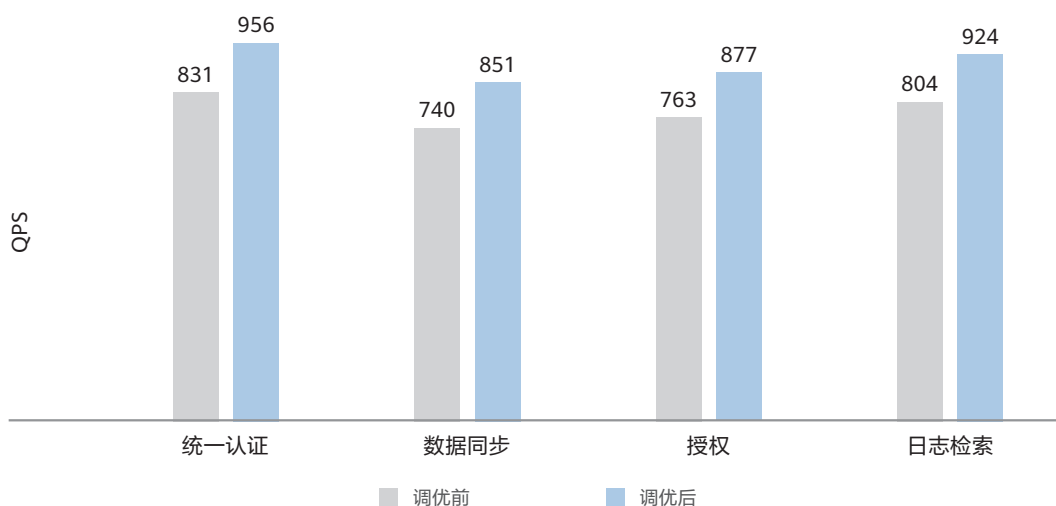
图 4-21 东华软件基于鲲鹏原生应用开发框架



### 4.5.2.3 业务价值

- » 原生高效率：改造后的流水线在整体开发、交付效率上提升约 25%
- » 原生高性能：HOS 门户系统的初期服务启动耗时从 60s 缩短至 45s，整体时间缩短 25%；门诊医生站、住院医生站、护士站等工作站加载时间降低约 25%；统一认证、数据同步、授权、日志检索等模块的每秒查询率 QPS 整体提升了 15%。

图 4-22 调优前后典型场景性能对比测试



东华软件的智慧医疗解决方案已在医疗信息化领域得到广泛应用，满足了医疗行业用户对于业务敏捷性、资源效率、系统复杂性管理、运维自动化以及技术多样性的迫切需求，实现了医疗信息系统的快速迭代、弹性伸缩以及高效运维。

未来，东华软件的医疗核心业务均将基于鲲鹏原生开发，促进整个医疗行业的技术创新与数字化转型，为患者提供更加高效、安全的医疗服务体验。

# 4.6 水平软件

## 4.6.1

### 星环科技

#### — 基于鲲鹏原生大数据基础软件，让企业数据流转更快捷

星环信息科技（上海）股份有限公司（简称：星环科技）是领先的企业级大数据和人工智能基础软件供应商，围绕数据的集成、存储、治理、建模、分析、挖掘和流通等数据全生命周期提供基础软件与服务。

#### 📄 4.6.1.1 业务背景

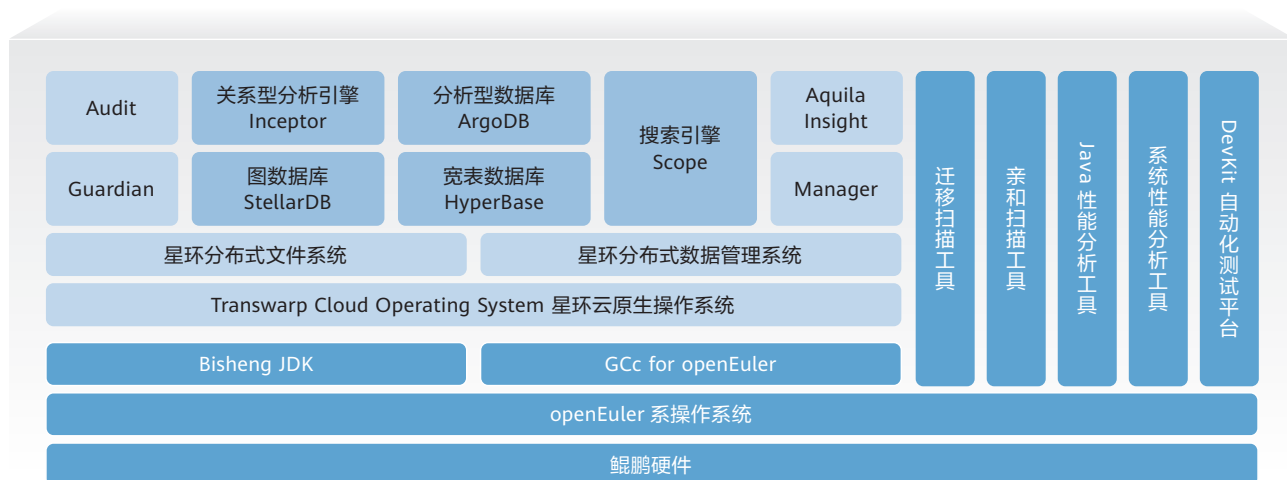
数字化时代，随着业务的快速发展，企业处理的不再是单一模态的数据，而是包含文本、图像、音频、视频等不同类型的非结构化数据、结构化数据、实时数据等，如何通过技术创新解决不同模态的数据处理、数据流转效率提升、业务响应时间缩短等问题是企业亟需解决的问题。

#### 🏠 4.6.1.2 鲲鹏原生方案

星环科技携手鲲鹏对五款核心产品进行鲲鹏原生优化和改造：星环科技大数据基础平台 TDH 中的搜索引擎 Scope、关系型分析引擎 Inceptor、图数据库 StellarDB、宽表数据库 Hyperbase，以及分布式分析型数据 ArgoDB。

在原有 CI/CD 平台中，增加本次五款产品的鲲鹏构建流程，基于鲲鹏服务器以及 openEuler 操作系统，搭建了持续集成节点；并将鲲鹏 DevKit 流水线技术栈（包括迁移扫描、亲和分析、毕昇编译器、鲲鹏测试框架、性能分析插件、版本漏洞扫描插件）融入自有的产品线管理、持续集成过程，形成新的构建集群，全面优化在鲲鹏架构上的开发及运行效率，进一步提升版本的性能、测试效率和质量。

图 4-23 星环大数据基础平台 TDH 原生开发架构图



代码开发阶段，使用鲲鹏 DevKit 快速创建场景化工程；在流水线构建过程中直接调用鲲鹏 DevKit 门禁检查组件，对代码的鲲鹏兼容性和鲲鹏架构亲和性进行检查和优化；在编译构建阶段，通过脚本的配置，无感知自动从开源 GCC 切换至 GCC for openEuler；构建完成后，在流水线中自动完成鲲鹏版本的调优、测试和发布，发布时通过增量 / 版本级编码扫描能力，对编码规范、病毒进行扫描，进一步保障版本的质量。

### 💎 4.6.1.3 业务价值

- » 原生高效率：经对比分析，基于改造后的流水线进行新版本开发，平均用时从 3 人日缩短至 2 人日，效率提升约 30%；
- » 原生高性能：通过替换开源 GCC 为 GCC for OpenEuler，以 ArgoDB 产品为例，产品代码编译构建效率提高了 27%；通过 Java 性能测试工具分析智能合约运行容器 JVM 中内存占用和 GC 回收等瓶颈问题，实现整体性能的提升 20%。

图 4-24 鲲鹏原生开发方案开发效率及性能优化对比



经过一系列优化，星环科技五款产品综合性能均得到明显提升，能够精准高效的处理企业结构化和非结构化数据的链上计算需求，让企业数据处理更加便捷和高效。

## 4.6.2

### 深信服

— 联合鲲鹏共同推进企业级分布式存储 EDS 原生开发创新实践，让数据管理更可靠

深信服科技股份有限公司（以下简称“深信服”）是一家专注于企业级网络安全、云计算、IT 基础设施与物联网的产品和服务供应商，承载十万多家行业用户数字化转型过程中的基石性工作，致力于让每个用户的数字化更简单、更安全。

#### 4.6.2.1 业务背景

数据存储的对企业数字化转型至关重要，随着大数据、云计算、人工智能等技术的飞速发展，数据量激增、数据类型更加丰富，如何构建一个高性能、高可靠、易扩展的数据存储底座，成为企业数字化转型过程中的关键课题。

#### 4.6.2.2 鲲鹏原生方案

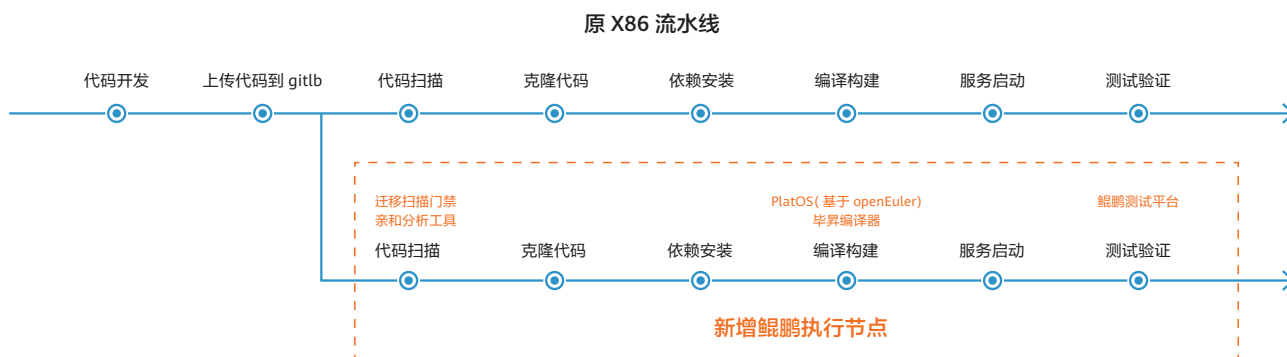
深信服首选核心产品企业级分布式存储 EDS V5.0.3，基于鲲鹏服务器 + openEuler 操作系统 + 鲲鹏 DevKit + 鲲鹏 BoostKit 进行原生开发创新实践。该产品是深信服的新一代软件定义存储解决方案，底层引入鲲鹏服务器、openEuler 操作系统和硬盘组建可横向扩展的存储资源池，在这个基础之上提供块存储、文件存储，包括对象存储，通过替换毕昇编译器、引入鲲鹏 BoostKit 加速算法进一步提升产品性能，从而更好的承载上层的虚拟化数据库、云原生、AI、HPC 等行业应用。

图 4-25 分布式存储 EDS 的鲲鹏原生架构图



研发人员将鲲鹏执行节点加入到现有流水线中，通过流水线脚本控制使用 x86 节点和鲲鹏节点，进行两个版本的应用构建，不影响现有流水线的工作内容；同时在鲲鹏执行节点中集成鲲鹏 DevKit 代码扫描门禁、亲和性分析门禁，安装鲲鹏高性能编译器、鲲鹏测试框架等，改造后的流水线已经应用到了深信服日常版本开发过程中，实现一套代码持续构建多架构版本，而且鲲鹏的软件版本出包的时候就是最优状态。

图 4-26 改造后的流水线

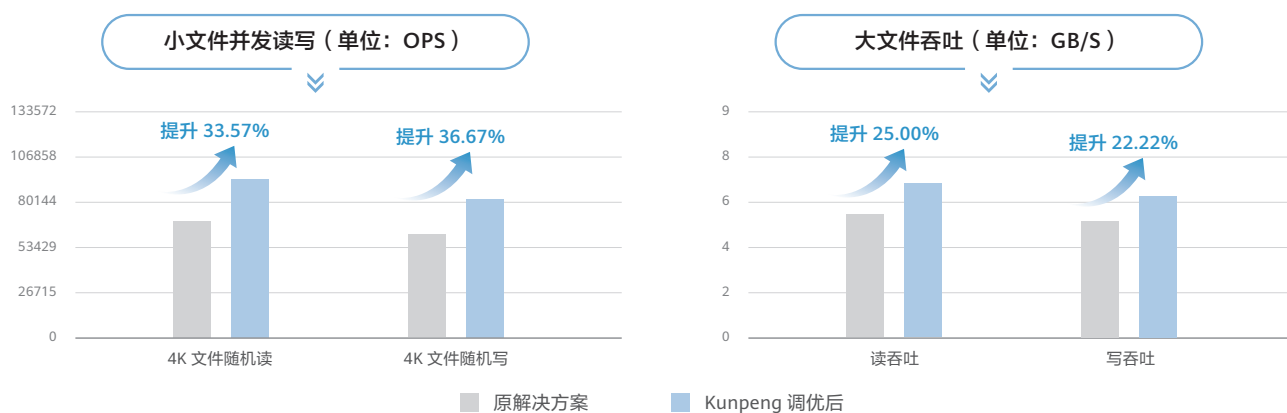


为了全面优化产品性能，EDS 存储团队联合生态使能团队结合鲲鹏软硬件开展全栈深度优化实践。通过 DevKit 系统性能分析快速识别出热点函数，并基于优化建议进行了修改；其次，使用鲲鹏 DevKit 对元数据访问进行全景分析，识别全路径 I/O 不同阶段时延损耗；针对损耗高环节进行优化，使用鲲鹏 BoostKit 提供的 SPDK 加速算法实现 NVME SSD 的高效访问，降低 IO 时延。

### 4.6.2.3 业务价值

原生高性能：经过上述一系列优化，深信服企业级分布式存储 EDS 性能显著提升，并发场景下小文件随机读写性能提升 35%，大文件吞吐型业务性能提升 25%；

图 4-27 鲲鹏优化前后性能对比



对数字化转型时代下数据类型的日益丰富和数据量的爆发式增长，深信服将持续深化鲲鹏原生创新实践，包括进一步优化流水线的原生开发能力、在算法特性层面对产品的深度优化、以及安全等新特性的增强等方面，打造更高效、更安全的数据存储与管理解决方案，助力企业数字化转型。