



鲲鹏介绍



前言

- 本章主要介绍鲲鹏计算产业以及鲲鹏生态，华为鲲鹏处理器的架构，型号与规格，技术创新与应用场景，TaiShan200机架服务器和高密服务器，华为云鲲鹏云服务，鲲鹏计算平台的操作系统兼容性以及openEuler开源操作系统的发展和特性等内容。



目标

- 学完本课程后，您将能够：
 - 了解计算产业的发展趋势，鲲鹏计算产业以及鲲鹏生态的全景图；
 - 描述华为鲲鹏处理器产品的性能规格、技术创新；
 - 了解TaiShan 200机架服务器和TaiShan 200高密服务器；
 - 华为云鲲鹏云服务的全景，了解常用的鲲鹏云服务的特性；
 - 描述鲲鹏计算平台兼容的操作系统，了解openEuler操作系统的特点以及openEuler开源社区的作用；



目录

1. 鲲鹏生态介绍

- 计算产业发展趋势
- 鲲鹏计算产业概述
- 鲲鹏计算产业生态

2. 华为鲲鹏处理器介绍

3. TaiShan服务器介绍

4. 华为云鲲鹏云服务介绍

5. 鲲鹏操作系统介绍



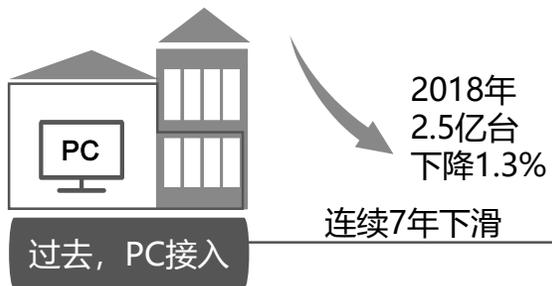
本节概述和学习目标

- 本节主要讲述计算产业发展趋势、鲲鹏计算产业概述以及鲲鹏计算产业生态。
- 学完本节后，您将能够：
 - 了解计算产业发展趋势
 - 了解鲲鹏计算产业组成
 - 了解鲲鹏计算产业生态

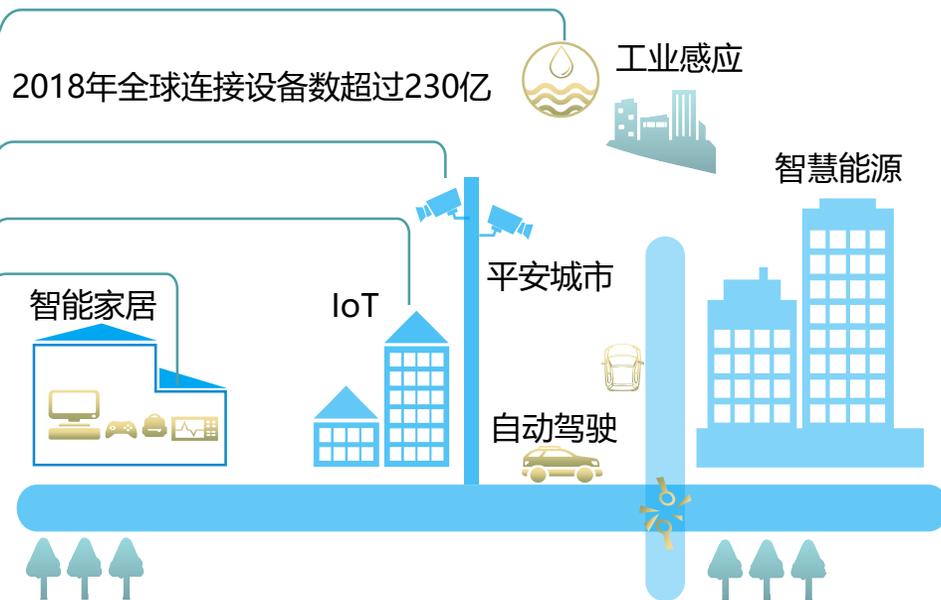


移动应用云化、万物互联兴起，呼唤新的算力

移动智能终端逐渐取代传统PC



世界正在进入万物互联的时代



传统PC向移动智能终端转移



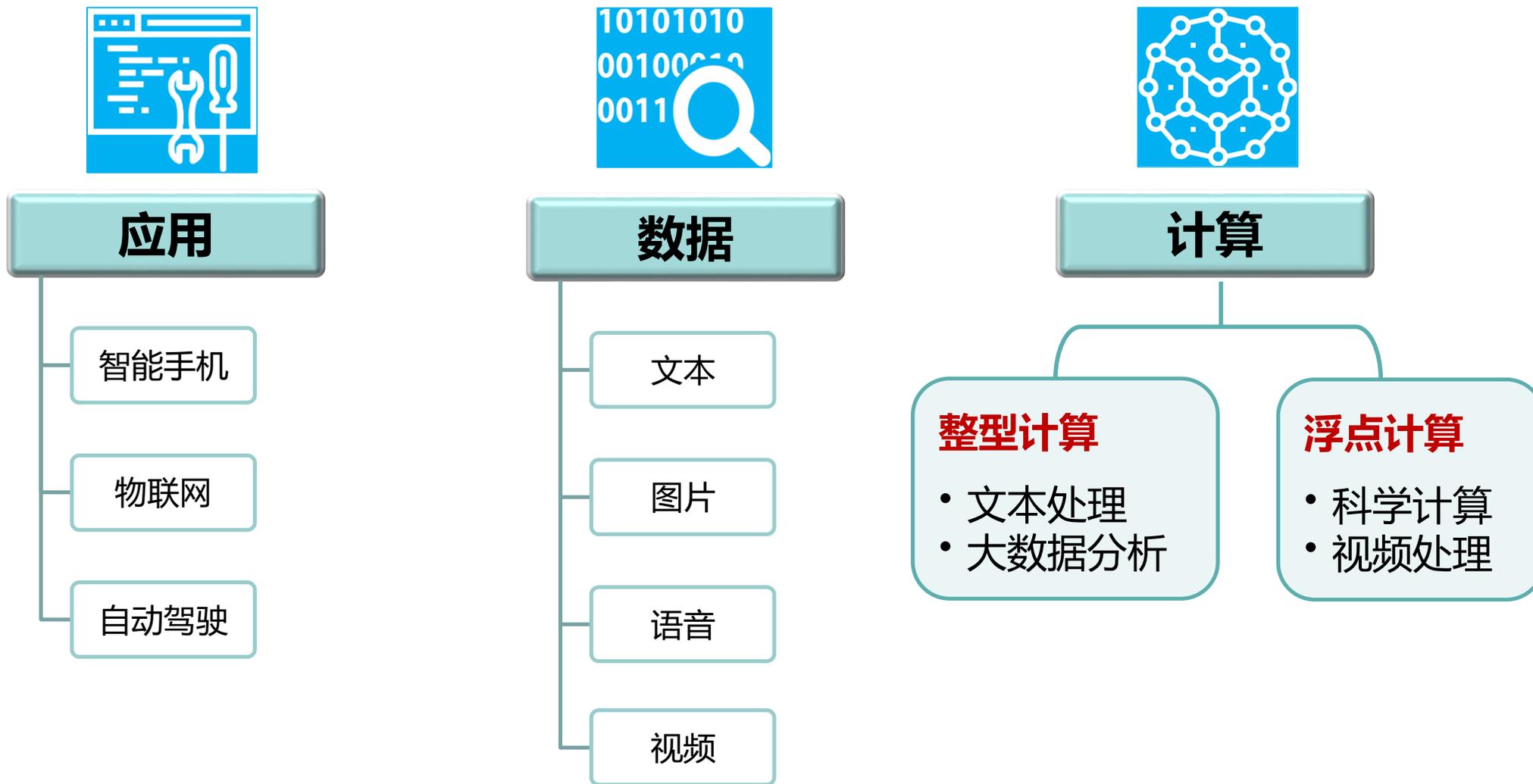
新的算力需求



海量数据产生



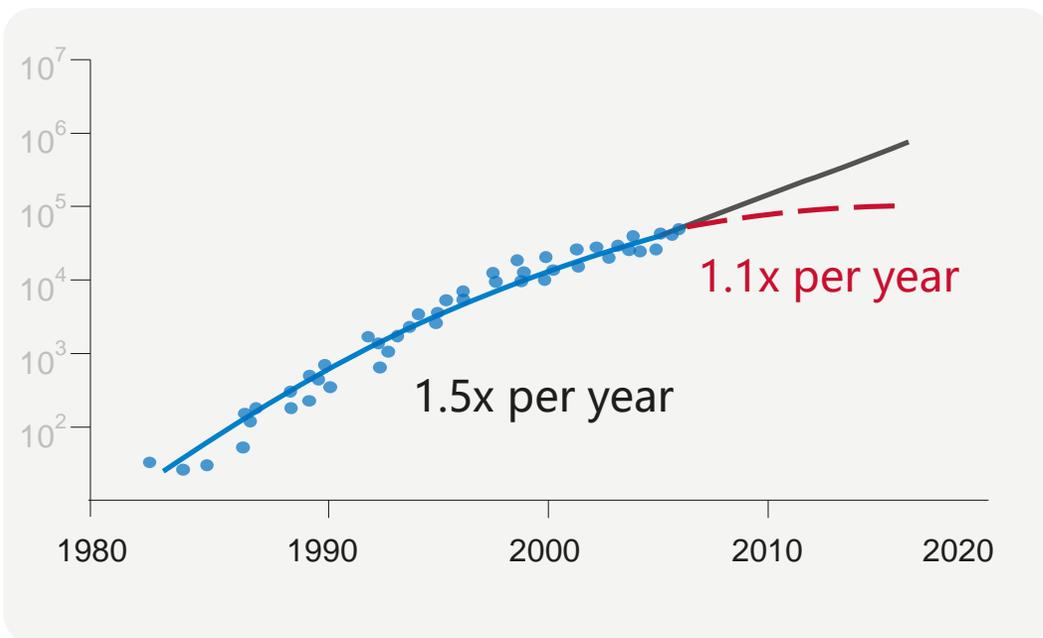
应用和数据的多样性需要新的计算架构



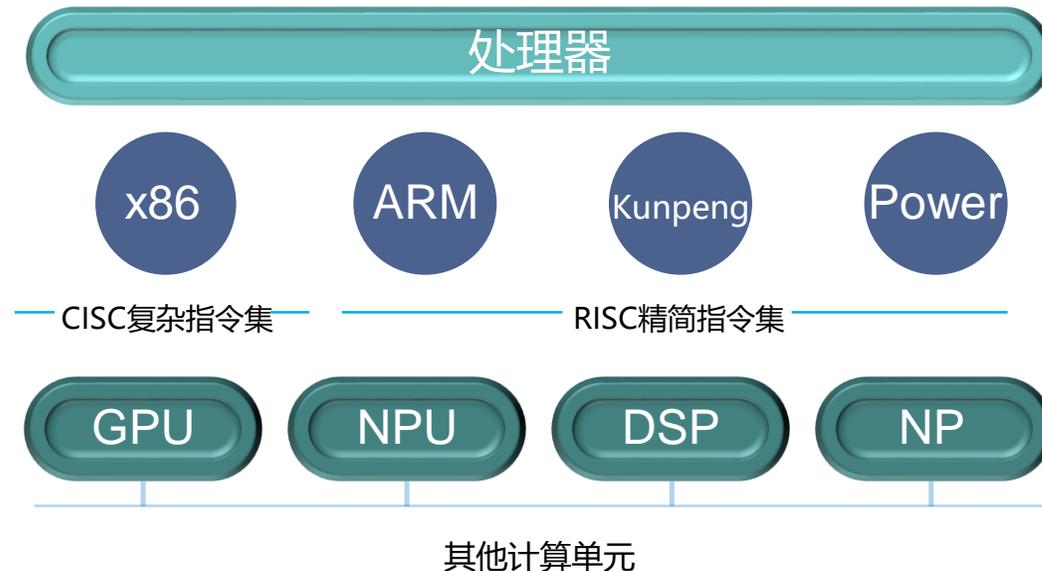


单一架构向多种计算架构组合演进

重构摩尔定律



多种计算架构并存的组合是最优的解决路径





超万亿规模的计算产业空间

- 新应用、新技术、新计算架构，百亿级联接、爆炸式数据增长将重塑ICT产业新格局，催生新的计算产业链条，涌现出新的厂家和
 - 软件：操作系统和虚拟化软件、数据库、中间件、大数据平台、企业应用软件，云服务、数据中心管理服务；
 - 硬件：服务器及部件、企业存储设备。



2023年全球计算产业投资额 (美元)



目录

1. 鲲鹏生态介绍

- 计算产业发展趋势
- 鲲鹏计算产业概述
- 鲲鹏计算产业生态

2. 华为鲲鹏处理器介绍

3. TaiShan服务器介绍

4. 华为云鲲鹏云服务介绍

5. 鲲鹏操作系统介绍



鲲鹏计算产业

- 鲲鹏计算产业是基于Kunpeng处理器构建的全栈IT基础设施、行业应用及服务，包括PC、服务器、存储、操作系统、中间件、虚拟化、数据库、云服务、行业应用以及咨询管理服务。

鲲鹏计算产业

- 行业应用
- 云服务
- 数据库
- 中间件
- 操作系统
- 虚拟化
- 存储 | PC
- 服务器
- Kunpeng 处理器



政府	金融	游戏	媒体与娱乐	运营商
鲲鹏ECS	鲲鹏BMS	鲲鹏容器	鲲鹏RDS	鲲鹏DWS
	兼容丰富的操作系统、中间件、数据库软件			
openEuler操作系统			高斯数据库	
TaiShan 2280 均衡型服务器	TaiShan 5280 存储型服务器	TaiShan X6000 高密型服务器	OceanStor V6/F V6 存储	
华为鲲鹏处理器	智能SSD控制器芯片	智能网卡芯片	智能管理芯片	



鲲鹏计算产业目标

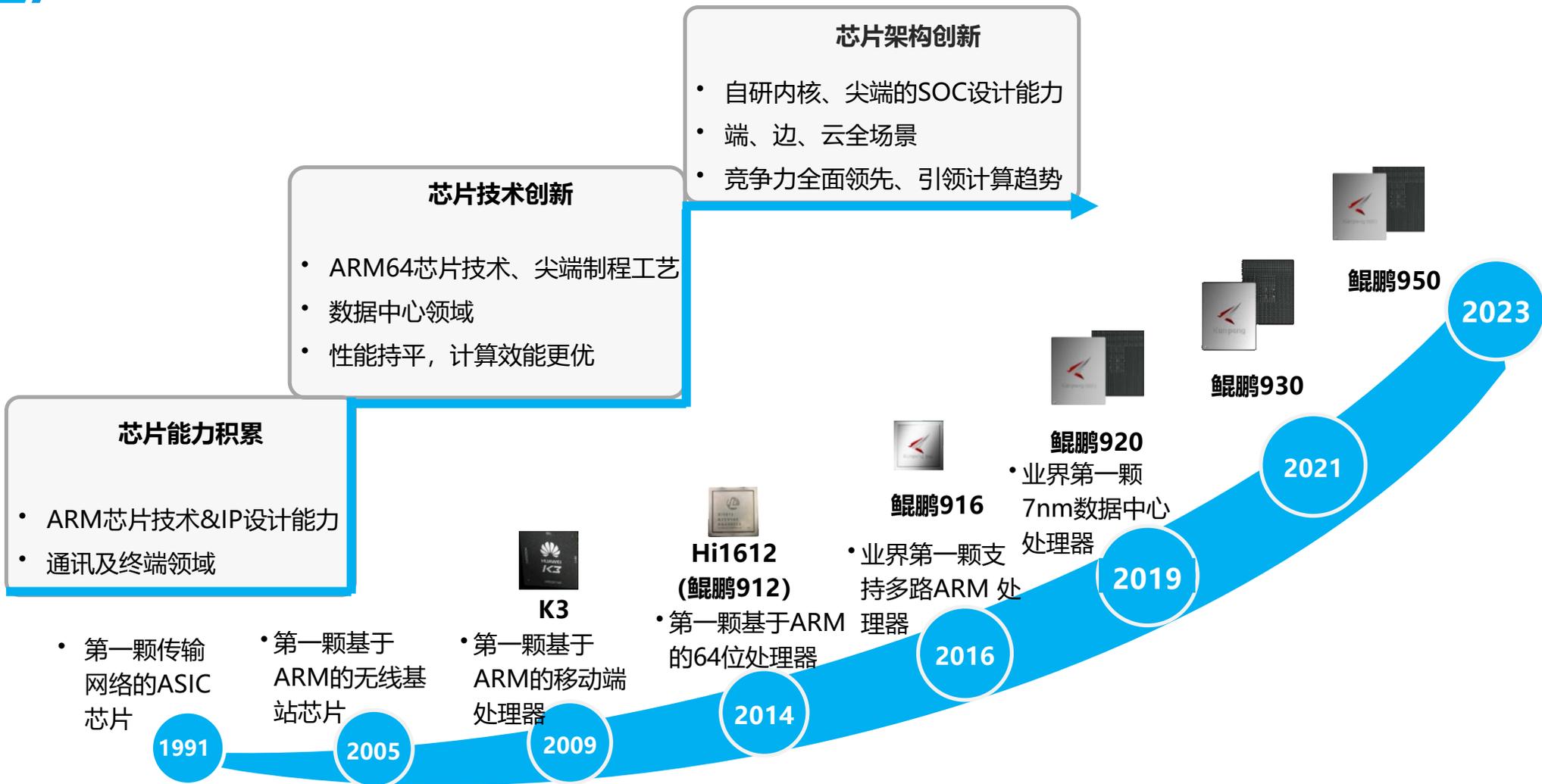
- 鲲鹏计算产业目标是建立完善的开发者和产业人才体系，通过产业联盟、开源社区、OpenLab、行业标准组织一起完善产业链，打通行业全栈，使鲲鹏生态成为开发者和用户的首选。

应用	支持迁移	免费支持主流应用和软件的迁移	
AI框架		软件 开源	开源AI框架（MindSore） 开源数据库（GaussDB OLTP单机版），使能伙伴发展自有品牌数据库产品 开源操作系统，使能伙伴发展自有品牌操作系统产品
中间件			
数据库			
操作系统			
整机	硬件 开放	提供Atlas模组、板卡，使能伙伴发展基于Atlas的智能化行业解决方案 提供主板，SSD/网卡/RAID卡等部件，使能伙伴发展自有品牌部件、服务器和PC等	
部件			
主板			
华为云	华为 聚焦	架构创新，处理器研发，云服务	
处理器			

● 华为 ● 合作伙伴



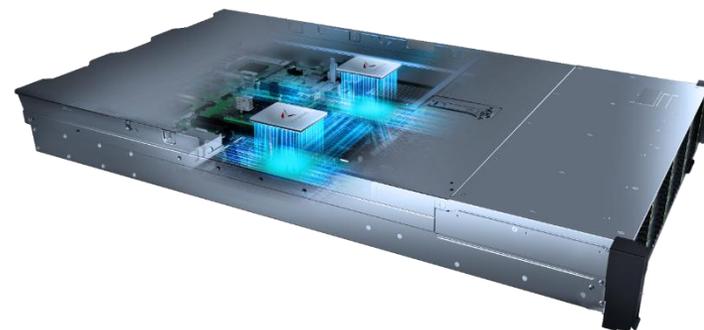
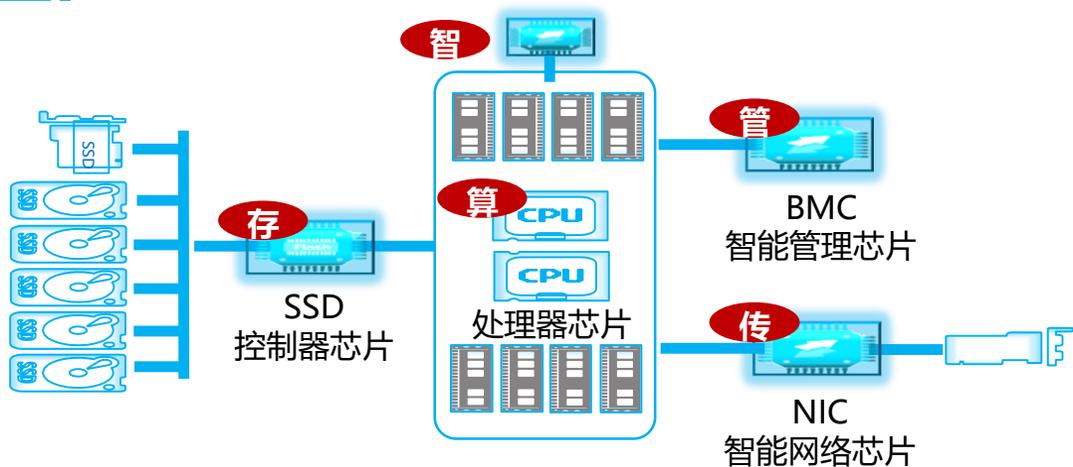
华为鲲鹏处理器，坚持持续创新



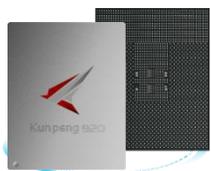


全面创“芯”，构建全自研服务器TaiShan

全自研服务器TaiShan



算



华为鲲鹏 920

ARM处理器芯片

首款7nm ARM服务器处理器，32/48/64核，2.6GHz

存



Hi1812

智能SSD控制芯片

PCIe NVMe与SAS融合智能加速，超强磨损算法

传



Hi1822

智能融合网络芯片

以太网与FC融合，协议加速，可编程

管



Hi1710

智能管理芯片

内置智能管理引擎
智能故障管理

AI



Ascend 310/910

人工智能芯片

达芬奇架构
极致能效与性能



华为云服务全面鲲鹏化，构建全栈鲲鹏云服务

华为鲲鹏EVS云盘



- 业界独家：首家基于ARM的云存储平台，率先提出云上All Flash战略
- 极致性能：业界最强云硬盘，时延低至50~100us，IOPS高达100万

华为鲲鹏数据库



- 高性价比：比同等能力x86的价格更便宜
- 应用透明：x86的应用切换到ARM架构下运行100%兼容
- 安全可信：支持更多加密算法，更安全可靠

华为鲲鹏Redis



- 业务首款：业界首款商用LibOS，超低时延
- 高性价比：价格相比行业优势低30%
- 轻量容器：极低内存开销

华为鲲鹏容器



- 业界独家：首发完整ARM容器平台
- 华为鲲鹏应用DevOps：提供完整的华为鲲鹏AI容器，价格低于业界30%

华为鲲鹏微服务平台



- 业界独家：首发ARM微服务引擎
- 华为鲲鹏微服务应用开发平台：提供微服务应用的编译、构建、部署、运行管理

华为鲲鹏应用运维



- 业界独家：首发ARM应用运维解决方案
- 华为鲲鹏应用运维管理平台：解决华为鲲鹏应用的问题定位和性能分析难题，改善应用体验



华为云鲲鹏云服务支持丰富场景

事物处理	大数据分析	数据库	科学计算	云服务	存储	移动原生应用
OLTP	OLAP	MySQL	CAE/CFD	前端Web	块存储	云游戏
Web Server	离线分析	Redis	CAD/EDA	数据Cache	对象存储	游戏开发测试
Email	AI-训练	Gbase	生命科学	搜索	文件存储	终端仿真
应用服务	AI-推理	达梦、东方通	分子动力学			移动办公
ERP		人大金仓	能源			可在ARM运行
CRM		Oracle	气象学			
		SAP	国防&安全			

- 各类业务中，开源软件可以在华为鲲鹏平台上运行，商业软件逐步完善
- 并行度较高或者可以并行化的应用，具有较好的性能



鲲鹏生态兼容的操作系统介绍

	社区发行版	商用发行版
国产 OS	 ... 	 中标麒麟 NeoKylin  银河麒麟 KYLIN 银河麒麟  deepin™  一铭软件 Emindsoft  BC-Linux  凝思 LINX-TECH  湖南麒麟 KYLINSEC ... 
国外 OS	 CentOS  debian ...	 SUSE® ...  ubuntu





鲲鹏计算产业优势



和ARM共享优势生态，协同加速发展

以中国市场孵化和完善行业应用，与全球产业形成良性循环



鲲鹏计算产业的典型应用

- 在5G、AI、云计算、大数据等技术推动下，各行各业对计算平台提出端云同构、海量多样化数据智能处理、实时分析等需求。Kunpeng处理器提供的强劲算力底座，将在各行业数字化转型过程中发挥重要作用。





目录

1. 鲲鹏生态介绍

- 计算产业发展趋势
- 鲲鹏计算产业概述
- 鲲鹏计算产业生态

2. 华为鲲鹏处理器介绍

3. TaiShan服务器介绍

4. 华为云鲲鹏云服务介绍

5. 鲲鹏操作系统介绍



鲲鹏计算产业生态全景

技术生态

鲲鹏计算平台是一个开放的技术生态，兼容业界主流操作系统，数据库，中间件等软件。



高校合作

校企合作，持续为计算产业培养和输出人才，合作方式：

- 校企联合课程
- 校企图书出版
- 高校实训室建设
- 人才双选会

开发者生态

鲲鹏计算平台鼓励开发者基于平台进行业务开发和创新：

- 鲲鹏开发者大赛
- 鲲鹏在线课程/云端实验室
- 鲲鹏职业认证

社区建设

鲲鹏社区为客户、伙伴、开发者提供丰富的资源和开放、平等的交流空间：



鲲鹏论坛二维码



产业生态

与伙伴和客户一道，共同打造面向行业的解决方案：



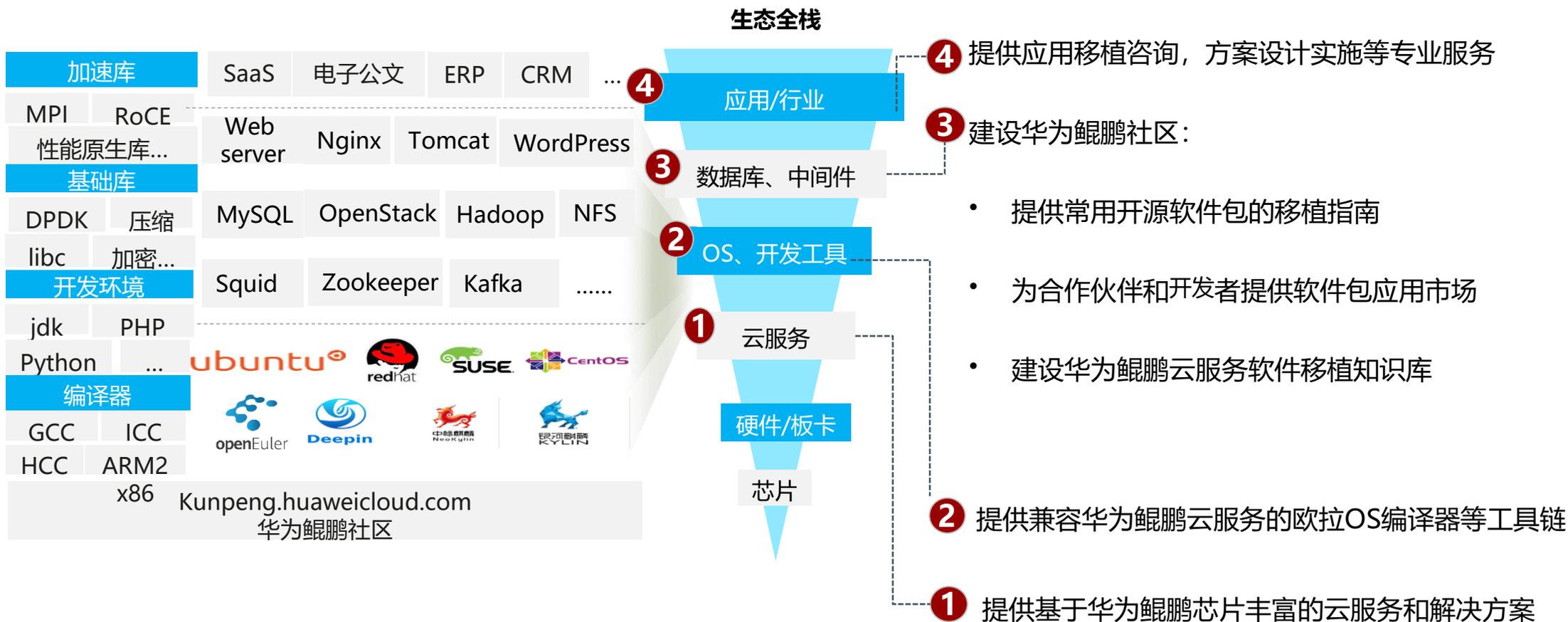
伙伴生态

三大伙伴计划为伙伴，提供培训、技术、营销、市场的全面支持：

- 鲲鹏凌云伙伴计划
- 鲲鹏展翅伙伴计划
- 鲲鹏智数伙伴计划



华为鲲鹏技术生态



注: 应用移植专业服务待上线



华为鲲鹏云专业服务

1. 咨询服务

- 迁移的可行性评估方案，包括软件兼容性评估、迁移工作量评估、风险评估等
- 迁移方案设计、包括工具和指导：ARM第三方仓库、编译移植工具、依赖lib包、最佳实践指导

2. 专家服务

- 针对不同服务包提供不同技术等级的技术支持工程师，提前预知技术风险并提供优化建议，指导或配合客户制定最佳华为鲲鹏云解决方案，为客户提供更全面，更贴身，更主动的技术服务

3. 实施服务

- 针对基础语言环境，运行环境，基础组件，数据库软件，组件安装后的问题定位、处理和跟踪，涉及开源软件的Bug反馈到社区并跟踪解决方案



服务形式: 远程服务，现场服务

4. 知识库服务

- 为客户提供基于华为云鲲鹏云的相关产品的单点镜像、集群环境下镜像组，及针对客户特定场景的镜像定制服务
- 为客户提供业务场景范围内的相关华为云鲲鹏云最佳实践及相关帮助文档

5. 售后服务

- 故障解决：非应用程序造成的故障定位、分析、应急解决方案制定、最终故障解决的服务
- 补丁升级：软件的漏洞预警、补丁评估、补丁验证以及补丁安装的服务
- 远程支持：400热线电话远程技术支持服务

6. 培训服务

- 为客户提供1-2次线上华为云鲲鹏云技术培训，包括：产品培训，技术培训，解决方案培训等



华为鲲鹏伙伴计划

伙伴计划

华为鲲鹏伙伴计划

子计划

华为鲲鹏凌云
伙伴计划

华为云服务鲲鹏
鹏子计划

华为鲲鹏展翅
伙伴计划

华为TaiShan服务器
鲲鹏子计划

华为鲲鹏智数
伙伴计划

华为智能数据&存储
鲲鹏子计划



华为鲲鹏伙伴计划公共权益

伙伴通过华为鲲鹏生态统一门户注册认证，并成功加入华为鲲鹏伙伴计划，可享受以下公共权益。

权益项	具体说明
培训支持	访问云学院，获得在线课程和在线实训
技术支持	应用移植开发文档工具包
	在线实验室/Openlab技术支持
营销支持	通过认证获得华为鲲鹏兼容性技术认证书
	使用华为鲲鹏伙伴名称及LOGO
业务支持	华为云测试券

各子计划权益及申请入口请上华为官网对应计划详情页了解：

鲲鹏凌云伙伴计划：<https://www.huaweicloud.com/partners/kunPeng/>

鲲鹏展翅伙伴计划：<http://ic-openlabs.huawei.com/#/teamdetail>

鲲鹏智数伙伴计划：<https://partner.huawei.com/>



鲲鹏社区



全栈资源

- 应用市场
- 代码仓库
- 编译器&工具链
- 交叉编译



知识经验

- 常用开源移植指南
- 社区和博客
- 华为鲲鹏论坛



人才圈子

- 基础课程
- 职业认证
- 大赛
- 人才培养



凌云计划

- 品牌合作
- 原厂认证
- 联合展示和宣传

Kunpeng.huaweicloud.com

华为鲲鹏社区



思考题

1. 华为鲲鹏计算产业相关产品有哪些？（ ）
 - A. 华为鲲鹏处理器
 - B. TaiShan服务器
 - C. 华为云鲲鹏云服务
2. 围绕鲲鹏计算产业，华为提供（ ）支持。
 - A. 云服务
 - B. 工具链
 - C. 社区服务
 - D. 专业服务



本节小结

通过本节的学习，需要掌握如下内容点：

- 计算产业发展趋势
- 鲲鹏计算产业组成
- 鲲鹏计算产业生态



目录

1. 鲲鹏生态介绍
2. **华为鲲鹏处理器介绍**
 - 华为鲲鹏处理器概述
 - 华为鲲鹏处理器架构介绍
 - 华为鲲鹏处理器型号及规格介绍
 - 华为鲲鹏处理器技术创新
 - 华为鲲鹏处理器应用场景介绍
3. Taishan服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍



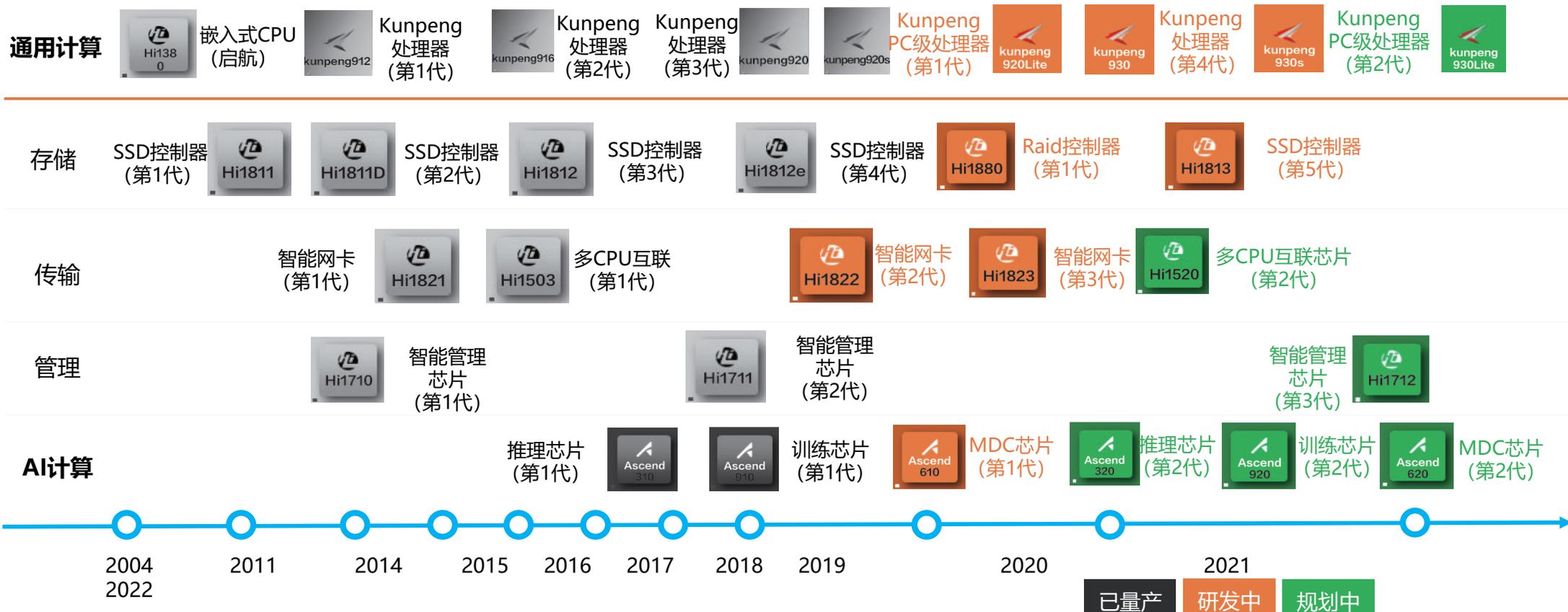
本节概述和学习目标

- 本节主要介绍华为鲲鹏处理器，包括其架构，型号及规格，技术创新以及使用场景等内容。
- 学完本课程后，您将能够：
 - 了解华为鲲鹏处理器的架构
 - 描述华为鲲鹏处理器性能规格
 - 描述华为鲲鹏处理器的技术创新
 - 描述华为鲲鹏处理器的特点和应用场景



华为鲲鹏处理器概述

- 华为鲲鹏处理器是华为自主研发的基于ARM架构的企业级系列处理器产品，包含“算、存、传、管、智”五个产品系统体系。





目录

1. 鲲鹏生态介绍
2. **华为鲲鹏处理器介绍**
 - 华为鲲鹏处理器概述
 - **华为鲲鹏处理器架构介绍**
 - 华为鲲鹏处理器型号及规格介绍
 - 华为鲲鹏处理器技术创新
 - 华为鲲鹏处理器应用场景介绍
3. Taishan服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍



华为鲲鹏处理器架构介绍

华为鲲鹏处理器基于ARM架构。ARM是一种CPU架构，有别于Intel、AMD CPU采用的CISC复杂指令集，ARM CPU采用RISC精简指令集（reduced instruction set computer，精简指令集计算机）。

	x86	ARM
指令集	CISC	RISC
供应商	主要有intel和AMD，Intel处于垄断地位	开放的授权策略，众多供应商
产业链	成熟	快速发展中



华为鲲鹏处理器架构（ARM）特点

优点

- 采用ARM架构，同样功能性能占用的芯片面积小、功耗低、集成度更高，更多的硬件CPU核具备更好的并发性能。
- 支持16位、32位、64位多种指令集，能很好的兼容从IOT、终端到云端的各类应用场景。
- 大量使用寄存器，大多数数据操作都在寄存器中完成，指令执行速度更快。
- 采用RISC指令集，指令长度固定，寻址方式灵活简单，执行效率高。

不足

- 在数据中心领域属于新进入者，其生态仍处于快速发展阶段。



目录

1. 鲲鹏生态介绍
- 2. 华为鲲鹏处理器介绍**
 - 华为鲲鹏处理器概述
 - 华为鲲鹏处理器架构介绍
 - 华为鲲鹏处理器型号及规格介绍
 - 华为鲲鹏处理器技术创新
 - 华为鲲鹏处理器应用场景介绍
3. Taishan服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍



华为鲲鹏通用计算处理器

华为从2004年开始基于ARM技术自研芯片，在通用计算处理器领域，2014年发布Kunpeng 912处理器，2016年发布Kunpeng 916处理器，2019年1月发布Kunpeng 920处理器，Kunpeng 920处理器是业界第一颗采用7nm工艺的数据中心级的ARM架构处理器。



**鲲鹏912
(Hi1612)**

第一颗基于
ARM的64位
CPU

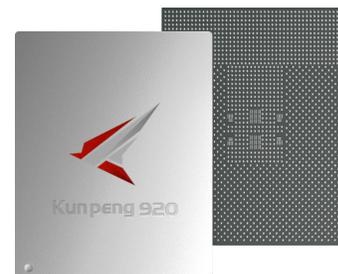
2014



鲲鹏916

业界第一颗支持
多路ARM CPU

2016



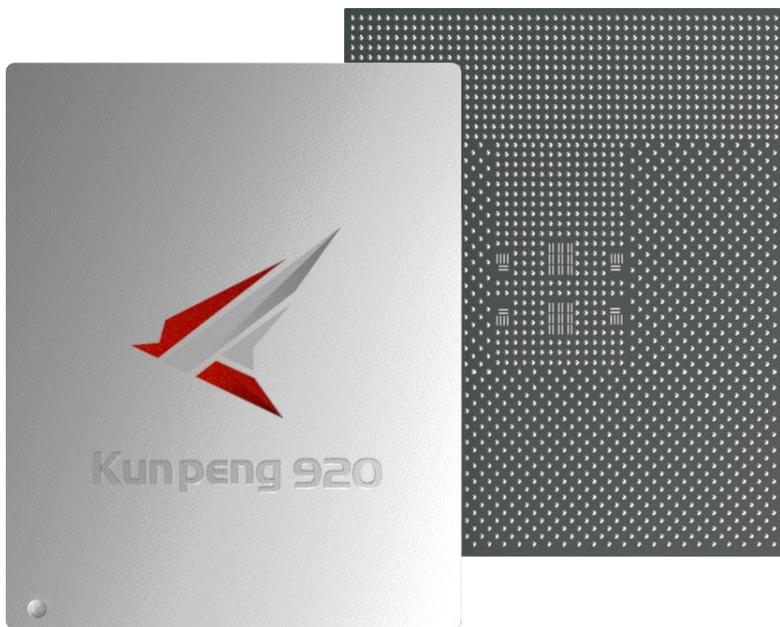
鲲鹏920

业界第一颗7nm数
据中心 ARM CPU

2019



华为鲲鹏920处理器规格



- 集成最多64*自研核，支持64核、48核、32核等多种型号
 - 指令集兼容ARMv8.2, 最高主频达2.6GHz
 - 每核集成64KB L1 I/D 缓存
 - 每核独享 512KB L2 缓存
 - 平均每核1MB L3 cache
- 8*DDR4控制器，最高可达2933MT/s
- 集成PCIe/SAS接口
 - 支持PCIe 4.0, 向下兼容PCIe 3.0/2.0/1.0
 - 支持x16,x8,x4,x2,x1 PCIe 4.0, 集成PCIe控制器
 - 支持16*SAS/SATA 3.0控制器
- 支持CCIX接口，支持加速器的缓存一致性
- 支持2*100G RoCE v2, 支持25GE/50GE/100GE标准NIC
- 支持2P/4P扩展
- 封装大小: 60mm*75mm



华为鲲鹏920处理器性能

高性能

930+, 25% ↑

Estimated SPECint_Rate_base2006 Score

高吞吐

内存带宽: 46% ↑
I/O 总带宽: 66% ↑
网络带宽: 4x

高集成

1 颗 = 4 颗芯片

高能效

30% ↑



与业界同期主流产品对比

*华为实验室测试数据，结果在不同环境中存在偏差



目录

1. 鲲鹏生态介绍
- 2. 华为鲲鹏处理器介绍**
 - 华为鲲鹏处理器概述
 - 华为鲲鹏处理器架构介绍
 - 华为鲲鹏处理器型号及规格介绍
 - 华为鲲鹏处理器技术创新
 - 华为鲲鹏处理器应用场景介绍
3. Taishan服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍



华为鲲鹏处理器技术创新



内核全自研

- 自研CPU内核，算力提升50%
- 自研片间支持2路/4路互联
- 自研多种硬件协加速引擎



内存/网络接口&IO协议

- 支持8通道DDR4内存控制器
- 支持100G RoCE端口
- 支持PCIe 4.0/CCIX协议



制程工艺领先

- 业界首款7纳米数据中心ARM CPU
- 采用业界领先的CoWoS封装技术，实现多Die合封



可靠性提升

- 支持使用概率高、收益大的RAS特性45条，实现在RAS特性上的增强



内核全自研，性能提升

高性能

Kunpeng 920 64核

Estimated SPECint®_rate_base2006 评估跑分

鲲鹏920-6426 (64核 2.6GHz) **930+**

Skylake Platinum 8180 750

高集成

Kunpeng 920



1 颗 = 4 颗芯片

高效能

Kunpeng 920 32/48核

Kunpeng-4826 **5.03**

Kunpeng-3226 4.52

Skylake Gold 6148 3.63

单位功耗的SPECint性能更高

低能耗

Kunpeng 916 32核

Skylake Gold 5115 0.31

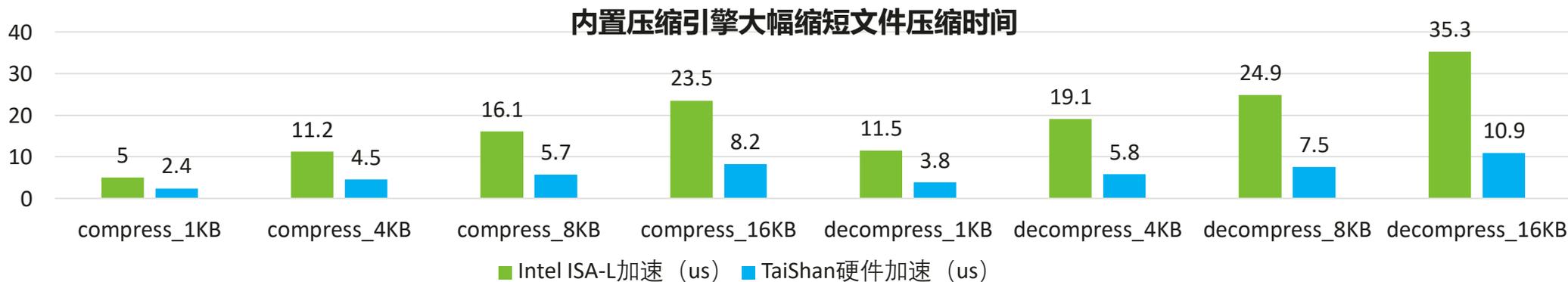
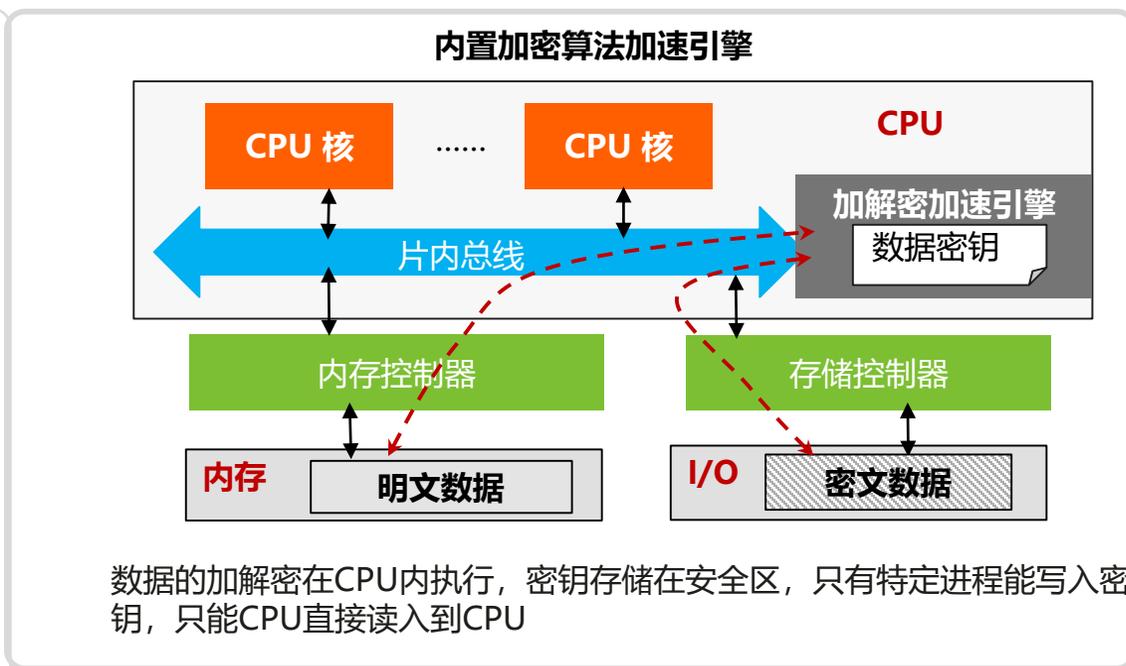
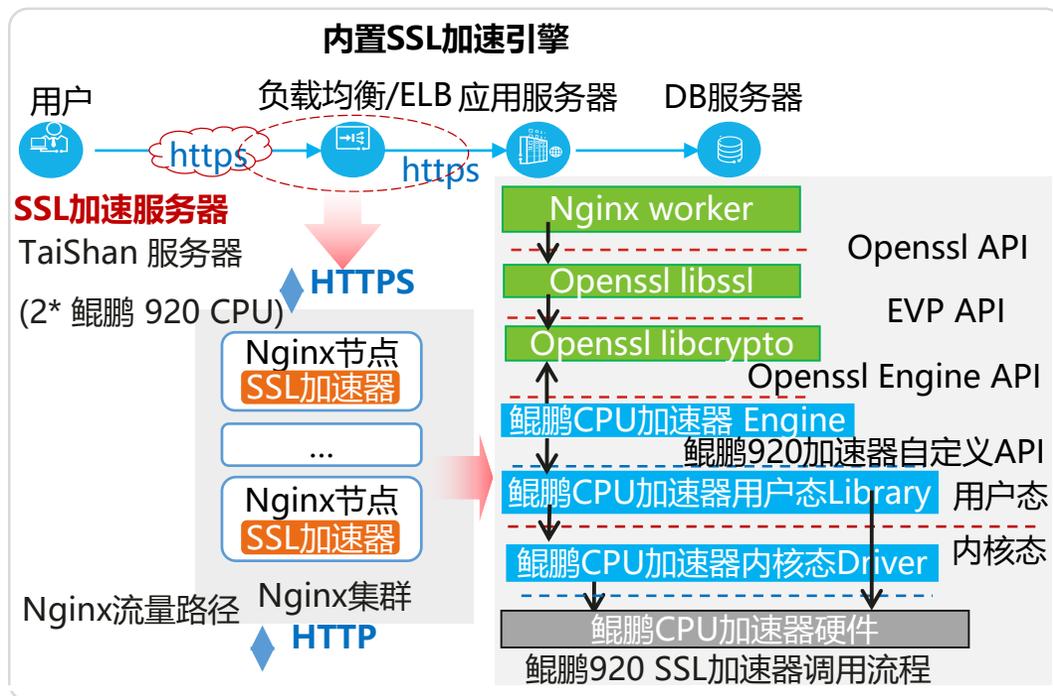
Kunpeng 916 0.25

单位SPECint性能的功耗更低

*华为实验室测试数据，结果在不同环境中存在偏差



Kunpeng 920内置多种加速引擎

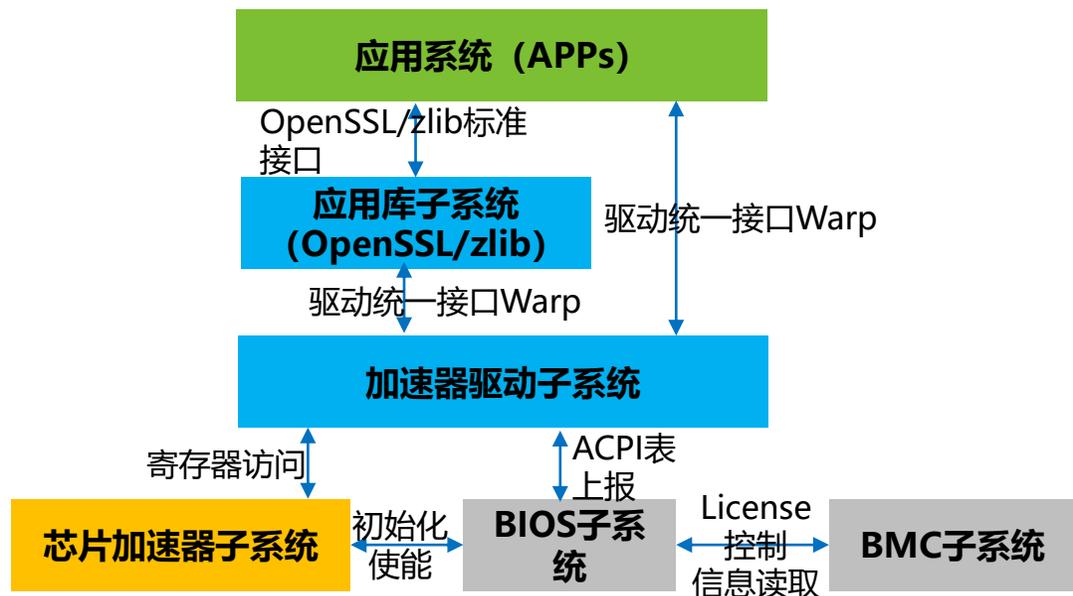




Kunpeng 920加速器简介

Kunpeng 920加速器系统逻辑架构

加速器子系统简介



目前加速引擎主要支持以下算法:

- 摘要算法SM3;
- 对称加密算法SM4, 支持CTR/CBC模式;
- 非对称算法RSA, 支持异步模型, 支持 Key Sizes 1024/2048/3072/4096;
- 压缩解压缩算法, 支持zlib/gzip.

安装方式: Kunpeng 920加速器子系统提供RPM安装和源码安装两种方式。



部署加速器环境要求与软件包下载

环境要求

- TaiShan 200服务器
- CPU: Kunpeng 920
- 操作系统*:
 - CentOS 7.6 4.14.0-115.el7a.0.1.aarch64 version
 - SUSE 15.1 4.12.14-195-default.aarch64 version
- 远程SSH登录工具已经在本地安装

*说明：RPM包目前只支持CentOS 7.6与SUSE 15.1两种操作系统，如需在其他操作系统使用加速库，建议使用源码方式编译安装。

通过华为support网站下载软件包和产品文档

版本文档				
资料名称	大小	发布时间	下载	
TaiShan Solution 2.0.0 加速器配套Openssl引擎 1.0.15 版本说明书 01	274.83(KB)	2019-08-06	↓	
TaiShan Solution 2.0.0 加速器配套Zlib库Patch 1.0.15 版本说明书 01	324.50(KB)	2019-08-26	↓	
TaiShan Solution 2.0.0 加速器驱动 1.0.15 版本说明书 01	278.54(KB)	2019-08-06	↓	
TaiShan Solution 2.0.0 版本配套表	139.48(KB)	2019-08-06	↓	
TaiShan Solution 2.0.0.SPC100. virus scan report	99.34(KB)	2019-08-06	↓	
TaiShan Solution 2.0.0.SPC100. virus scan report_update	99.34(KB)	2019-08-26	↓	

下载

软件				
软件名称	下载次数	大小	发布时间	下载
hisi_hpre-1.0.15-1.centos7.6.aarch64.rpm	1	150.62(KB)	2019-08-06	↓
hisi_hpre-1.0.15-1.sles15.1.aarch64.rpm	1	168.68(KB)	2019-08-06	↓
hisi_rde-1.0.15-1.centos7.6.aarch64.rpm	0	186.25(KB)	2019-08-06	↓
hisi_rde-1.0.15-1.sles15.1.aarch64.rpm	1	200.24(KB)	2019-08-06	↓
hisi_sec2-1.0.15-1.centos7.6.aarch64.rpm	1	160.97(KB)	2019-08-06	↓
hisi_sec2-1.0.15-1.sles15.1.aarch64.rpm	1	174.77(KB)	2019-08-06	↓
hisi_zip-1.0.15-1.centos7.6.aarch64.rpm	0	138.65(KB)	2019-08-06	↓



部署鲲鹏加速器 - 安装OpenSSL

OpenSSL安装过程

拷贝OpenSSL源码包

编译OpenSSL

设置环境变量

安装检查

说明：必须使用OpenSSL 1.1.1a或以上版本。Gcc推荐使用7.4.1及以上版本，make推荐使用3.82及以上版本。

步骤说明

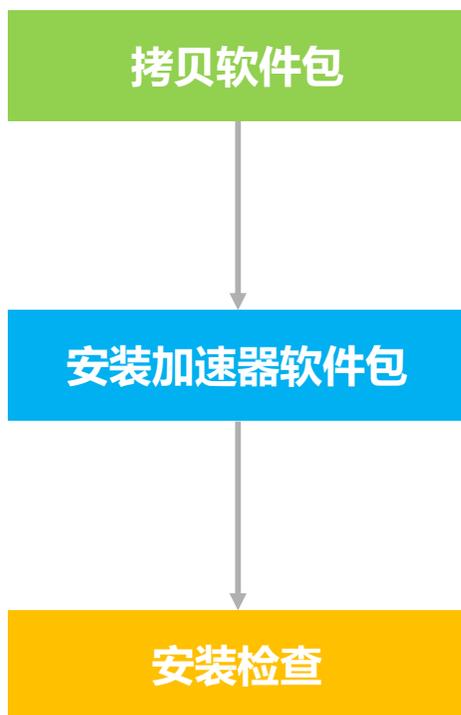
- 1、将OpenSSL源码包拷贝到安装目录下
- 2、编译OpenSSL
 - 2.1 执行./config生成makefile
 - 2.2 执行make命令编译OpenSSL
 - 2.3 执行安装脚本make install，默认部署路径为/usr/local
- 3、设置环境变量（以默认路径为例）

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export OPENSSL_ENGINES=/usr/local/lib/engines-1.1
```
- 4、安装检查
 1. 执行cd命令，进入到/usr/local/bin目录下
 2. 执行./openssl version命令查看版本信息
[root@localhost bin]# ./openssl version
OpenSSL 1.1.1a 20 Nov 2018 //显示该格式内容说明安装成功



安装加速器软件包（RPM包安装）

加速器软件包安装过程



说明：只支持CentOS 7.6与SUSE 15.1两种操作系统进行RPM方式安装，其他OS请按照源码方式安装。

步骤说明

1、将加速器软件包拷贝到安装目录下

1.1 安装前需完成OpenSSL 1.1.1a及以上版本已正确安装

2、安装加速器软件包

2.1 执行rpm -ivh命令安装加速驱动软件包，以uacce为例

```
[root@localhost home]# rpm -ivh uacce-1.0.1-1.centos7.6.aarch64.rpm
```

```
Preparing... #####  
[100%]
```

```
checking installed modules
```

```
Updating / installing...
```

```
1:uacce-1.0.1-1.centos7.6
```

```
#####
```

```
[100%]
```

```
modules installed
```

2.2 参考步骤2.1，依次安装hisi hpre, hisi_sec2, hisi_rde, hisi_zip驱动软件包和libwd, libkae引擎软件包。

3、安装检查

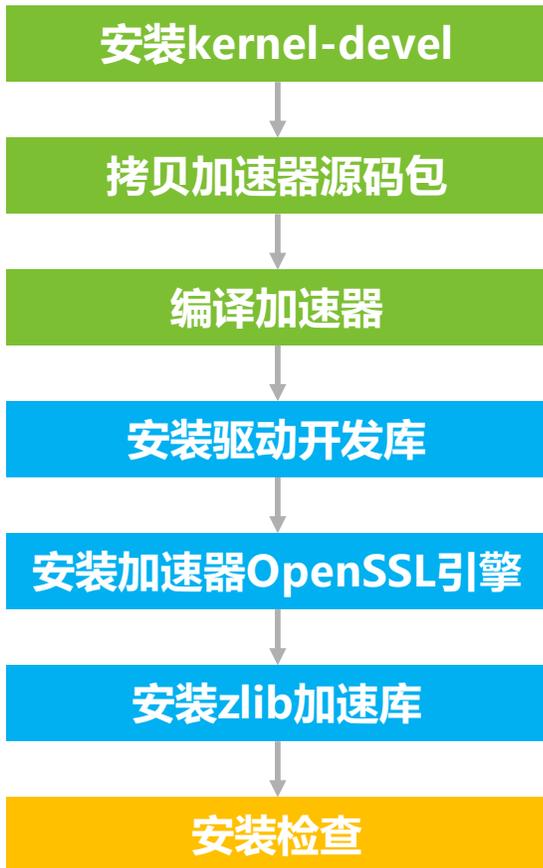
3.1 使用rpm -qa |grep 查看加速器软件包。

```
[root@localhost home]# rpm -qa uacce  
uacce-1.0.1-1.centos7.6.aarch64 //显示该格式内容说明安装成功
```



安装加速器软件包（源码安装）

加速器软件包安装过程



安装步骤说明

- 1、安装内核开发包kernel-devel
- 2、将加速器源码包拷贝到自定义目录
- 3、在下载好的源码目录下，使用make命令编译加速器
- 4、编译安装Warpdrive驱动开发库
- 5、编译安装加速器OpenSSL引擎
- 6、编译安装zlib加速库
- 7、进入到/usr/local/lib目录下，检查lib库软链接状态

说明：

- 1、安装前需确保OpenSSL 1.1.1a版本已经正确安装。
- 2、详细安装步骤请参考《TaiShan200服务器加速引擎开发者指南》



加速器测试

使用KAE引擎加速后签名性能从724.1 sign/s提升到2819sign/s

使用OpenSSL的软件算法测试RSA性能

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed rsa2048
...
sign    verify    sign/s  verify/s
rsa 2048 bits  0.001384s  0.000035s  724.1  28365.8.
```

使用KAE引擎的测试RSA性能

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae rsa2048
...
sign    verify    sign/s  verify/s
rsa 2048 bits  0.000355s  0.000022s  2819.0  45478.4
```

使用KAE引擎加速后异步RSA签名性能从735.7 sign/s提升到 54384.1 sign/s

使用OpenSSL的软件算法测试异步RSA性能

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -async_jobs 36 rsa2048
...
          sign    verify    sign/s  verify/s
Rsa 2048 bits  0.001318s  0.000032s  735.7  28555
```

使用KAE引擎的测试异步RSA性能

```
linux-rmw4:/usr/local/bin # ./openssl speed -engine kae -elapsed -async_jobs 36
rsa2048
....
          sign    verify    sign/s  verify/s
rsa 2048 bits  0.000018s  0.000009s  54384.1  105317.0
```

使用KAE加速后SM4 CBC模式在102400块大小下从每3秒处理2196次提升到11409次

使用OpenSSL的软件算法测试SM4 CBC模式性能

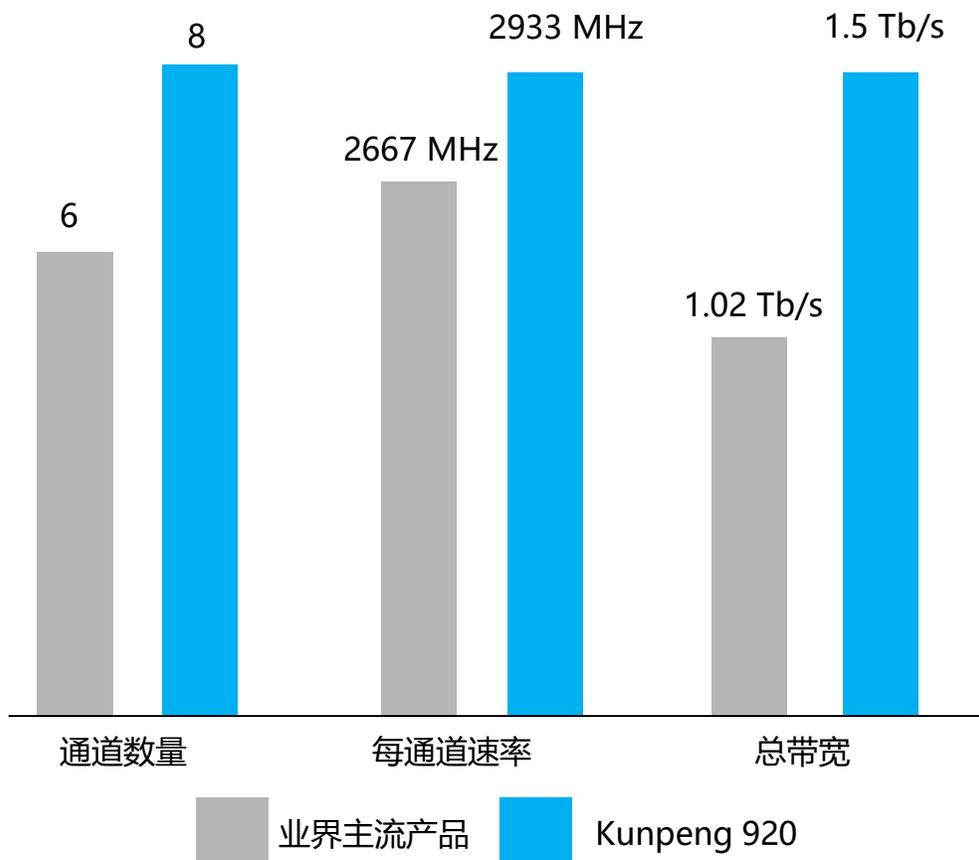
```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp sm4-cbc
You have chosen to measure elapsed time instead of user CPU time.
....
Doing sm4-cbc for 3s on 10240 size blocks: 2196 sm4-cbc's in
3.00s ....
type      51200 bytes 102400 bytes 1048576 bytes 2097152 bytes 4194304
bytes 8388608 bytes
sm4-cbc 82312.53k  85196.80k    85284.18k   85000.85k
85284.18k    85261.26k
```

使用KAE引擎的测试SM4 CBC模式性能

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae -evp sm4-cbc
engine "kae" set.
You have chosen to measure elapsed time instead of user CPU time.
...
Doing sm4-cbc for 3s on 1048576 size blocks: 11409 sm4-cbc's in 3.00s
...
Type      51200 bytes 102400 bytes 1048576 bytes 2097152 bytes 4194304
bytes 8388608 bytes
sm4-cbc 383317.33k  389427.20k  395313.15k  392954.73k  394264.58k
394264.58k
```



支持8通道DDR4内存控制器

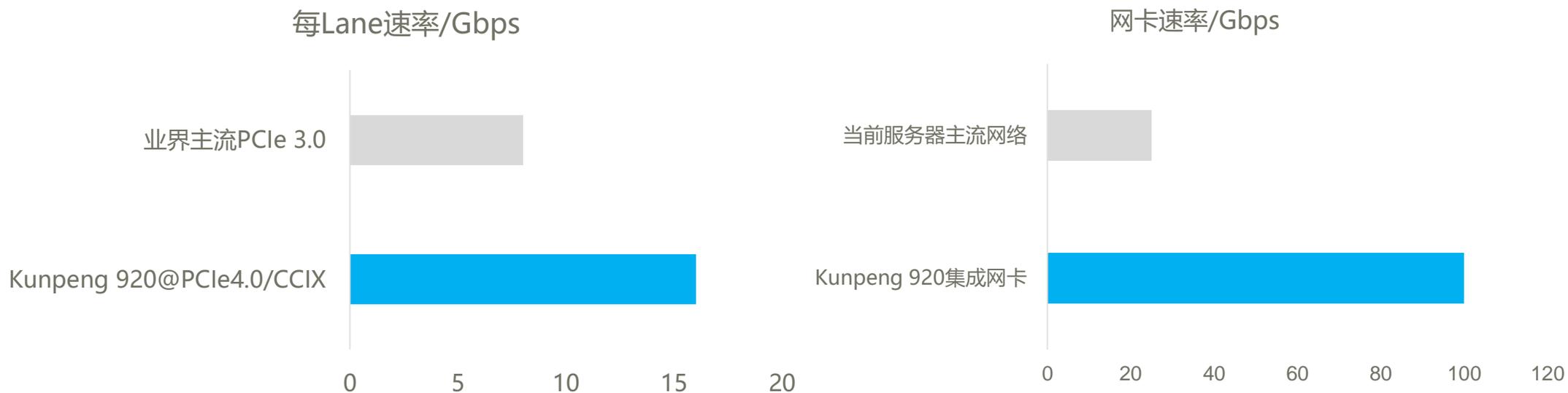


- 8通道DDR4 带来46% 带宽提升，同时容量也可按需提升
- 延迟优化，和业界主流水平相当/更优
- Load to use latency
 - 9 cycle to local L2
 - 25~33 cycle to local partition L3
 - 47~ 82 cycle to remote partition L3
- 完整的Cache & Memory QoS 方案，为用户的不同业务部署带来方便



网络端口&IO协议

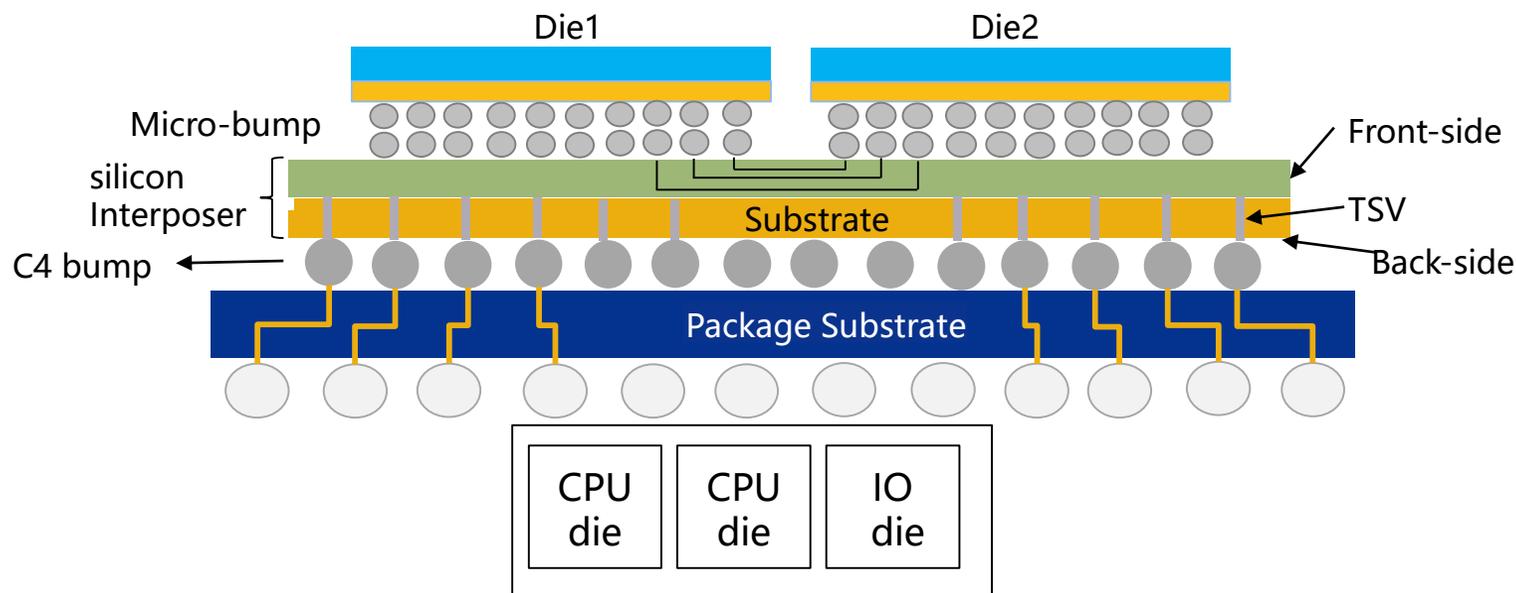
Kunpeng 920 集成了PCIe4.0/CCIX, 以及集成了100GE RoCEv2 (延迟和同等带宽IB相当), 为客户的网络/存储/加速卡等外围接口带来更高的效率。





制程工艺领先

- Kunpeng 920处理器，采用业界领先的7nm工艺，是首款采用7nm工艺制作的数据中心ARM CPU。
- 采用业界领先的CoWoS封装技术，实现多Die合封，控制每Die面积，提升良率，降低整体成本，乐高方式更加灵活。



华为鲲鹏920芯片



可靠性提升

华为鲲鹏920芯片结合IT产品实际应用需要，新增使用概率高，收益大的RAS特性45条，其中具有代表性的特性如下：

类别	特性
内存	可纠正错误精确统计
	自适应多区域替换 (ADDDC)
PCIE	错误在线恢复 (OER)
CPU	Core的错误容忍 (PDC)
系统	错误恢复
	可纠正错误阈值和计数



目录

1. 鲲鹏生态介绍
- 2. 华为鲲鹏处理器介绍**
 - 华为鲲鹏处理器概述
 - 华为鲲鹏处理器架构介绍
 - 华为鲲鹏处理器型号及规格介绍
 - 华为鲲鹏处理器技术创新
 - 华为鲲鹏处理器应用场景介绍
3. Taishan服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍



ARM架构处理器应用领域

目前超过90%的移动终端采用的是ARM架构的处理器。

随着IOT、AI和业务云化的发展，ARM在终端的优势地位将会带动其进入数据中心市场，成为下一个快速增长的市场领域。





华为鲲鹏处理器的应用场景

分布式存储



多核架构适合分布式存储软件

- 软件管理面与数据面分离，分别绑定到足够的CPU核心上，避免相互干扰；
- 软件进程绑定CPU核心，减少操作系统的调度抖动；
- 更多核可设置更多独立队列，降低噪声；

大数据



多核高并发架构匹配大数据负载应用特征

- 采用ARM架构，更多的物理核数可以提高大数据应用任务并发度，获得更好的性能；

原生应用



与移动端同构，上云效率高

- 移动端超过90%应用基于ARM架构原生开发；
- 采用ARM服务器平台具备同构优势，二进制直接迁移上云；
- 应用100%兼容；

原生BYOD



基于移动瘦终端的移动办公，信息更安全

- 更多的移动终端平台；
- 更直观的在线预览；
- 更完美的协同功能；
- 更安全的数据保护；

游戏云



典型业务上云场景，创造新的商业模式

- 应用预览：无需下载即可体验，即用即抛；
- 社交体验：游戏分享，多人协同；
- 用户数据产生价值：精准广告推送，无内购订阅模式提升忠诚度；
- 应用开发更加高效灵活：一次开发支持安卓和iOS，快速上线推广；



思考题

1. 以下哪些关于华为鲲鹏920处理器的描述是正确的？（ ）
 - A.采用了7nm的制造工艺；
 - B.支持8通道的DDR4控制器；
 - C.支持PCIe 4.0接口，并兼容PCIe 3.0/2.0/1.0；
 - D.支持多种加速器；
2. 华为鲲鹏920处理器内置了那些加速器？（ ）
 - A. SSL加速引擎
 - B. 加解密加速引擎
 - C. 压缩解压缩加速引擎



本节小结

- 本章介绍了华为鲲鹏处理器架构、型号及特性及应用场景，重点介绍了华为鲲鹏处理器的技术创新，帮助学员了解和掌握华为鲲鹏处理器的相关知识。



目录

1. 鲲鹏生态介绍
2. 华为鲲鹏处理器介绍
- 3. Taishan服务器介绍**
 - TaiShan服务器概述
 - TaiShan 200机架服务器产品介绍
 - TaiShan 200机架服务器价值特性
 - TaiShan 200高密服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍

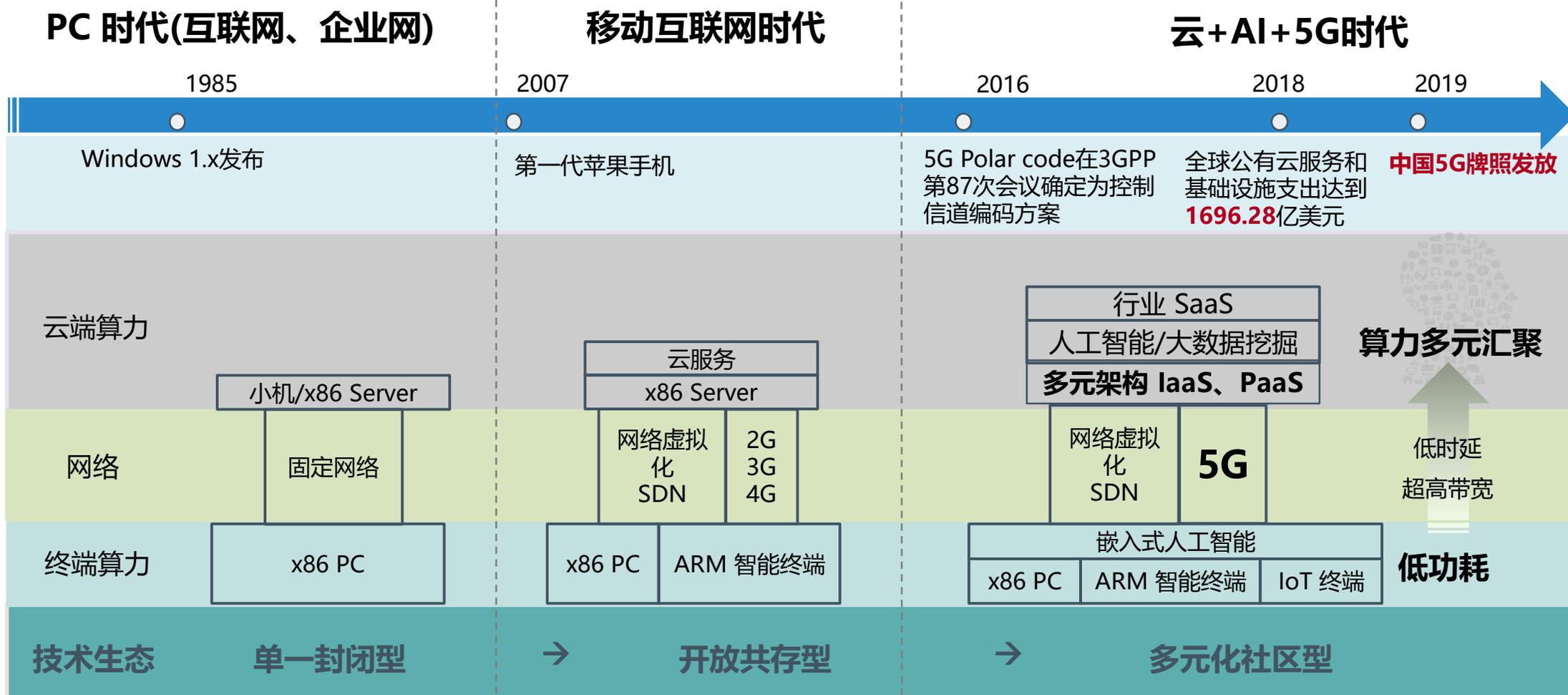


本节概述和学习目标

- 本节主要介绍TaiShan服务器的概述、型号以及技术创新等内容。
- 学完本节课程后，您将能够：
 - 描述TaiShan服务器的概况
 - 掌握TaiShan 200机架服务器的型号及规格
 - 掌握TaiShan 200机架服务器的技术创新
 - 掌握TaiShan 200高密服务器的型号及特性



云+AI+5G时代，多元架构成为业务智能的关键





基于华为鲲鹏处理器，构建整机计算能力

高效能计算

提供兼容ARM架构的高性能华为鲲鹏处理器、TaiShan服务器和解决方案，将高效能计算带入数据中心；

安全可靠

华为鲲鹏处理器基于自研内核，TaiShan服务器计算芯片全自研；17年计算创新铸就稳如泰山的高品质；

开放生态

开放平台，支持业界主流软硬件；构建鲲鹏生态，与开发者、伙伴和产业组织共同打造智能计算新底座；





TaiShan服务器系列介绍

TaiShan 100服务器

- 基于Kunpeng 916处理器
- 最多16个DDR4内存
- 支持5个PCIe 3.0扩展插槽
- 支持SAS/SATA硬盘和SSD
- 支持板载GE/10GE网络



TaiShan 200服务器

- 基于Kunpeng 920处理器
- 最多32个DDR4内存
- 支持最多8个PCIe 4.0扩展插槽
- 支持NVMe SSD、SAS/SATA硬盘和SSD
- 支持100GE板载网络
- 支持板级和全液冷技术



目录

1. 鲲鹏生态介绍
2. 华为鲲鹏处理器介绍
3. **Taishan服务器介绍**
 - TaiShan服务器概述
 - TaiShan 200机架服务器产品介绍
 - TaiShan 200机架服务器价值特性
 - TaiShan 200高密服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍



TaiShan 200服务器全景图



华为TaiShan服务器

17年的工程工艺积累

板级液冷散热设计

无源背板&三重硬盘抗震

散热液冷

高速互联

可靠设计

质量品控

56G板级高速互联

故障率低于业界15%

数据中心

存储密集型

5290
4U72盘存储型



(新产品)

5280
4U40盘存储型



2280
2U2路均衡型



1280
1U2路高密型



(新产品)

2480
2U4路高性能型



(新产品)

计算密集型

X6000
2U4节点高密型



边缘计算

2280E
边缘型



(新产品)

1280和2480计划于2019年Q4上市；5290和2280E计划于2020年Q1上市



TaiShan 200机架服务器全景图

2U2路



产品型号

2280

价值定位

均衡型

上市时间

2019H1

4U2路



5280

存储型

2019H1

2U4路



2480

高性能型

2020Q1

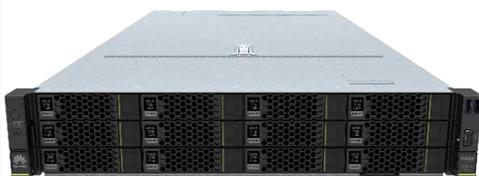




2280均衡型规格与亮点

2280 均衡型

前视图



后视图

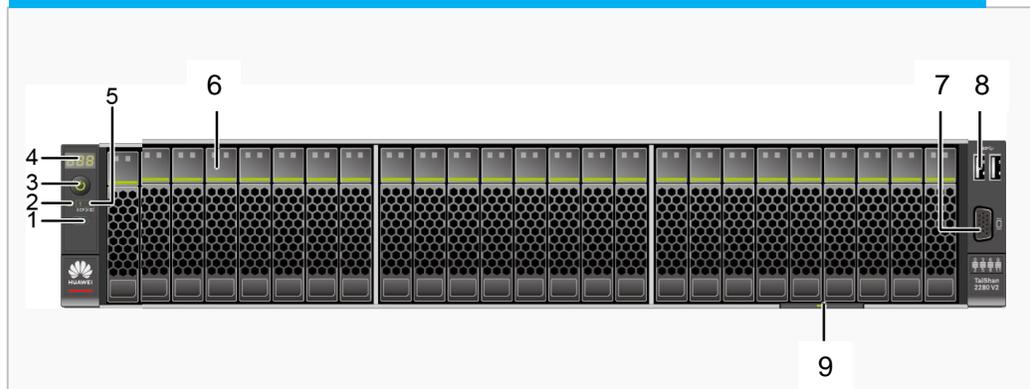
图无法显示此图像。

型号	2280	主要特点
形态	2U2路	高性能 <ul style="list-style-type: none"> 鲲鹏920处理器，性能比肩x86高端型号 8通道内存技术，支持32个DDR4内存插槽，最高内存容量可达4TB 支持华为Atlas 300 AI加速卡、ES3000 V5 NVMe SSD 灵活适配 <ul style="list-style-type: none"> 支持多个IO模组，实现丰富的硬盘配置 支持板载的灵活网卡，支持GE/10GE/25GE，实现不同网络配置， 安全可靠 <ul style="list-style-type: none"> 采用华为全自研计算芯片 整机器件实现全国产化，保障可持续供应
CPU	2*鲲鹏920处理器，可支持32,48,64核可选，主频2.6GHz，最高支持180W	
内存	32个DDR4 DIMM，最高2933MT/s，最大内存4TB	
本地存储	<ul style="list-style-type: none"> 16*3.5" SAS/SATA HDD硬盘 27*2.5" SAS/SATA HDD硬盘 16*3.5" NVMe SSD硬盘 	
RAID支持	RAID0/1/10/1E/5/50/6/60等	
PCIe扩展	最多8个PCIe 4.0 x8或3个PCIe 4.0 x16 + 2个PCIe 4.0 x8标准插槽	
板载网卡	4*GE OR 4*10GE OR 4*25GE	
电源	2个900W或2000W热插拔电源，支持AC 220V或DC 240V，支持1+1冗余	
风扇	4个热拔插风扇，支持N+1冗余	
温度范围	5°C~40°C	



2280型号前后面板

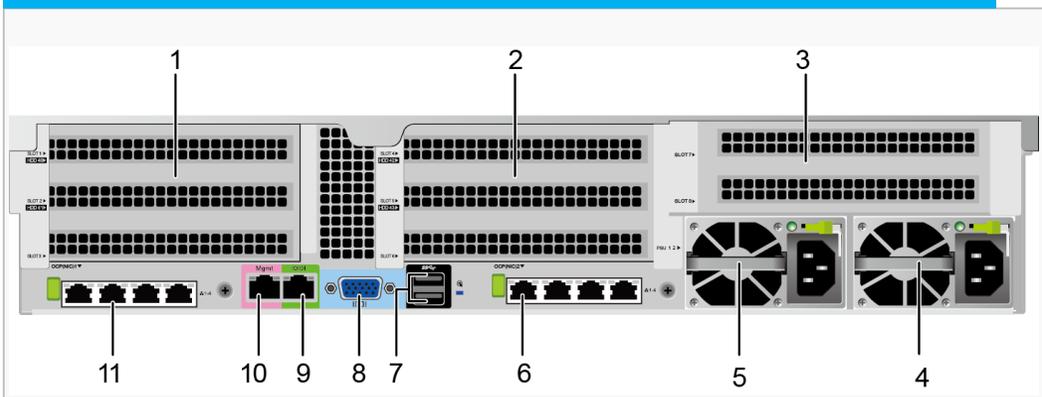
前面板



指示及按钮

按钮	指示	按钮	指示
1	UID按钮/指示灯	2	健康状态指示灯
3	电源开关按钮/指示灯	4	故障诊断数码管
5	灵活IO卡在位指示灯 (1, 2)	6	硬盘-
7	VGA	8	USB 3.0
9	标签卡		

后面板



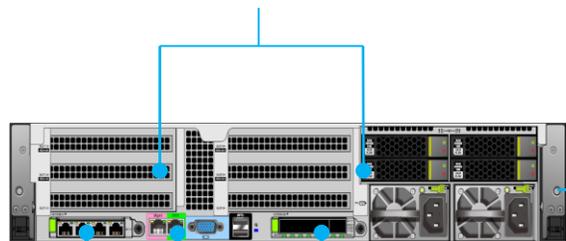
指示及按钮

按钮	指示	按钮	指示
1	IO模组1	2	IO模组2
3	IO模组3	4	电源模块2
5	电源模块1	6	灵活IO卡2
7	USB 3.0接口	8	VGA接口
9	调试串口	10	管理网口
11	灵活IO卡1	-	-



2280型号内部结构

6*PCIe4.0 x8全高槽位,
或可配置为2组
2*3.5"/2.5" SAS/SATA/SSD硬盘



2*GE,串口,
管理网口

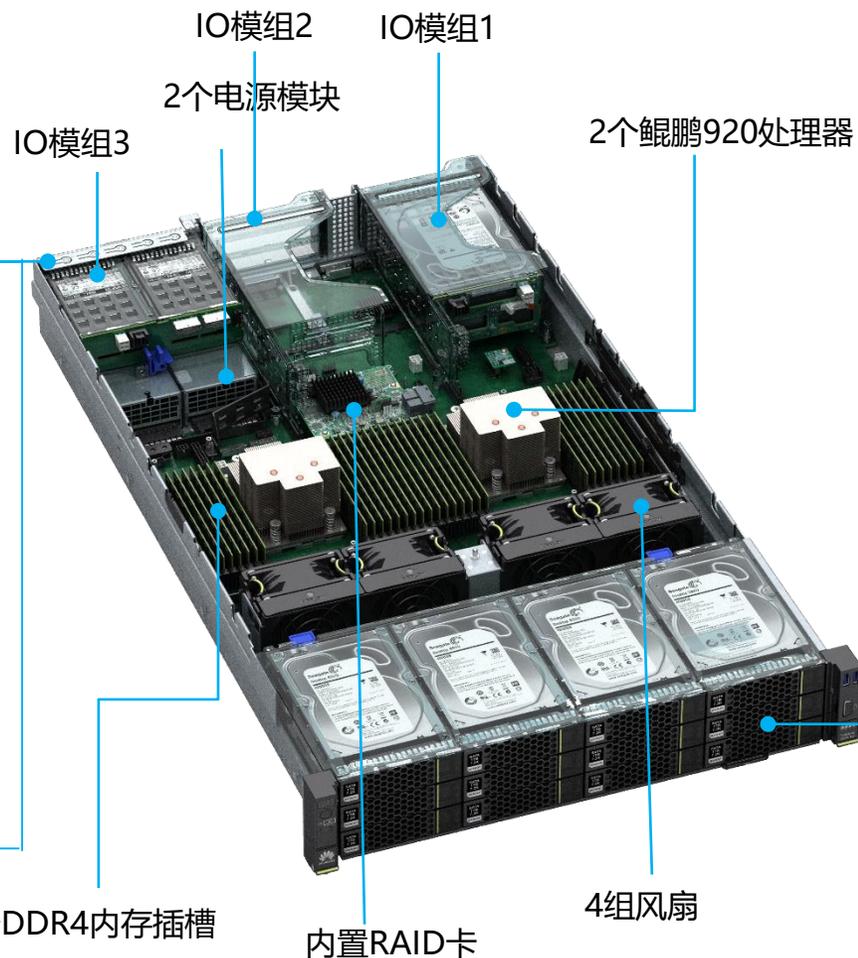
灵活网卡1

灵活网卡2

后置4*2.5" SAS/SATA/SSD/NVMe
SSD硬盘, 或可配置为2*PCIe4.0 x8
全高半长槽位



32个DDR4内存插槽



硬盘框A: 12*3.5" SAS/SATA硬盘



硬盘框B: 25*2.5" SAS/SATA硬盘



硬盘框C: 8*2.5" SAS/SATA+12NVMe硬盘





2280型号机型配置

类别	前置	后置	RAID方案
1 12盘位硬RAID方案	 12*3.5" SAS/SATA硬盘	 4*3.5" SAS/SATA硬盘 4*2.5" NVMe硬盘	 RAID卡
2 12盘位SAS直通方案	 12*3.5" SAS/SATA硬盘	 2*2.5" SAS/SATA硬盘 4*2.5" NVMe硬盘	CPU SAS直通+软RAID
3 25盘位通用硬盘方案	 25*2.5" SAS/SATA硬盘	 2*2.5" SAS/SATA/硬盘 4*2.5" NVMe硬盘	 RAID卡
4 25盘位NVMe SSD方案	 8*2.5" SAS+12*2.5" NVME硬盘	 4*2.5" NVMe硬盘	 RAID卡



5280存储型规格与亮点

5280 存储型

前视图



后视图



型号	5280	主要特点
形态	4U	超大存储 <ul style="list-style-type: none"> 支持最多40个3.5" 硬盘，本地存储容量可达560TB 鲲鹏920处理器，性能比肩x86高端型号，高效支持数据存储 8通道内存技术，支持32个DDR4内存插槽，最高内存容量可达4TB 灵活适配 <ul style="list-style-type: none"> 支持多个IO模组，实现丰富的硬盘配置 支持板载的灵活网卡，支持GE/10GE/25GE，实现不同网络配置， 安全可靠 <ul style="list-style-type: none"> 采用华为全自研计算芯片 整机器件实现全国产化，保障可持续供应
CPU	2*鲲鹏920处理器，可支持32,48,64核可选，2.6 GHz频率可选；最高支持200W	
内存	32个DDR4 DIMM插槽，最高2933MT/s	
本地存储	<ul style="list-style-type: none"> 36/40*3.5" SAS/SATA HDD硬盘 4*2.5" NVMe SSD硬盘 	
RAID支持	RAID0/1/10/1E/5/50/6/60等	
PCIe扩展	最多8个PCIe 4.0 x8或3个PCIe 4.0 x16 + 2个PCIe 4.0 x8标准插槽	
灵活网卡	4*GE OR 4*10GE OR 4*25GE	
电源	2个2000W 热插拔电源，支持1+1冗余	
风扇	4个热插拔风扇，支持N+1冗余	
温度范围	5°C~35°C	



5280型号前后面板

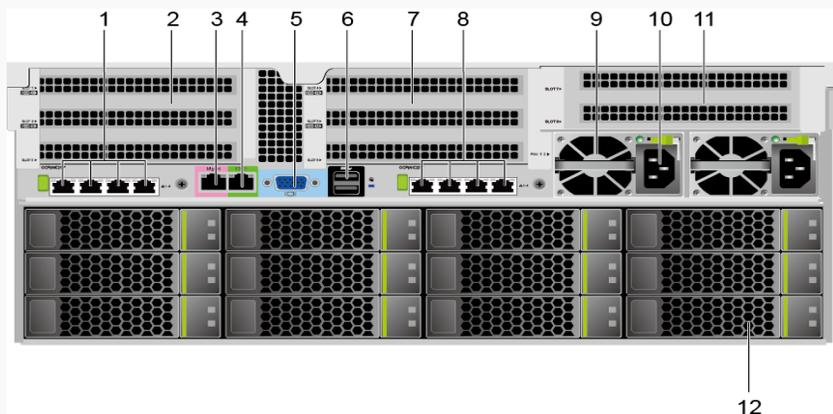
前面板



指示及按钮

按钮	指示	按钮	指示
1	UID按钮/指示灯	2	健康状态指示灯
3	电源开关按钮/指示灯	4	故障诊断数码管
5	灵活IO卡在位指示灯 (1, 2)	6	硬盘-
7	VGA	8	USB 3.0
9	标签卡		

后面板

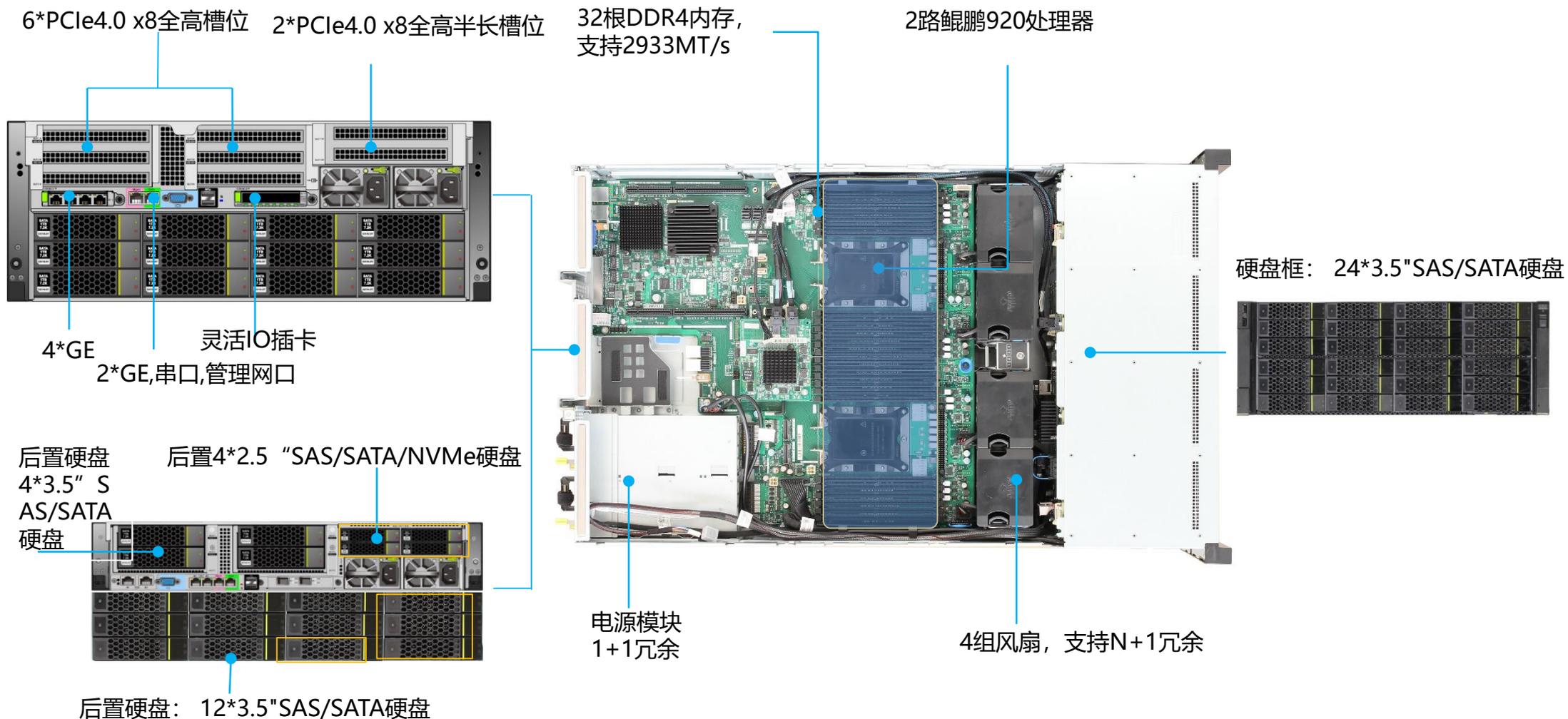


指示及按钮

按钮	指示	按钮	指示
1	灵活IO卡1	2	IO模组1
3	管理网口	4	调试串口
5	VGA接口	6	USB 3.0接口
7	IO模组2	8	灵活IO卡2
9	电源模块	10	电源模块接口
11	IO模组3	12	硬盘



5280型号内部结构





目录

1. 鲲鹏生态介绍
2. 华为鲲鹏处理器介绍
- 3. Taishan服务器介绍**
 - TaiShan服务器概述
 - TaiShan 200机架服务器产品介绍
 - TaiShan 200机架服务器价值特性
 - TaiShan 200高密服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍



TaiShan 200机架服务器特点



2280均衡型



5280存储型

高性能



超强算力: 高性能Kunpeng 920处理器

大内存容量: 8通道内存技术, 支持32个DDR4内存插槽

超强AI加速: 支持华为Atlas 300 AI加速卡, 满足多场景推理

灵活适配



灵活的IO扩展: 支持灵活插卡和标准的智能网卡, 实现丰富的网络配置

分级存储: 支持大容量存储硬盘和ES3000 V5 NVMe SSD

安全可靠



安全、可供应: 采用华为全自研计算芯片, 整机器件全国产化

可靠与质量保障: 减振、散热等高可靠设计; 质量流程与测试



超强算力：Kunpeng 920处理器

高性能

930+ 3x ↑
Estimated SPECint®_rate_base2006 评估跑分

高吞吐

内存带宽: **2.4x ↑**
I/O 总带宽: **1.7x ↑**
网络带宽: **10x ↑**

高集成

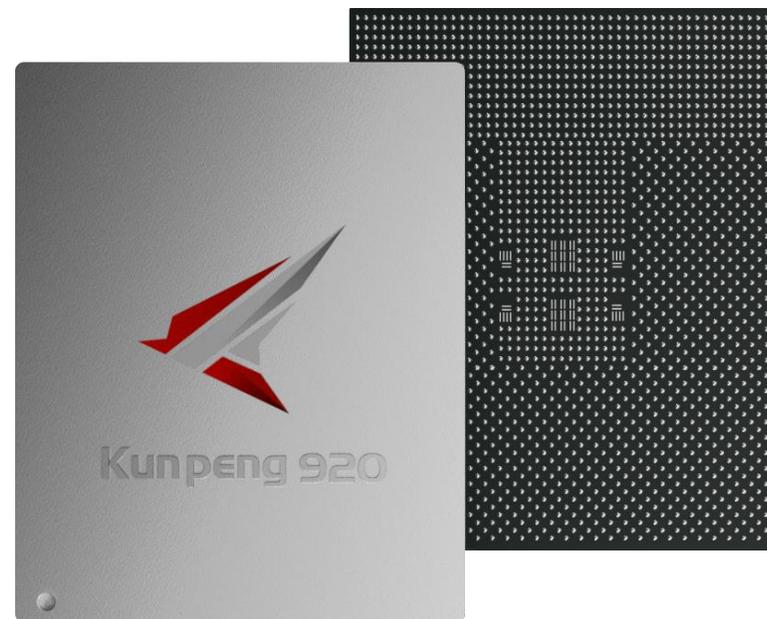
1 颗 = 4 颗芯片
(CPU, 南桥、网卡、SAS控制器)

高效能

35% ↑

* 基于Kunpeng 920-6426处理器 vs Kunpeng 16处理器的对比数据

* 对比来源华为实验室测试数据，结果在不同环境中可能有偏差



64核

8通道

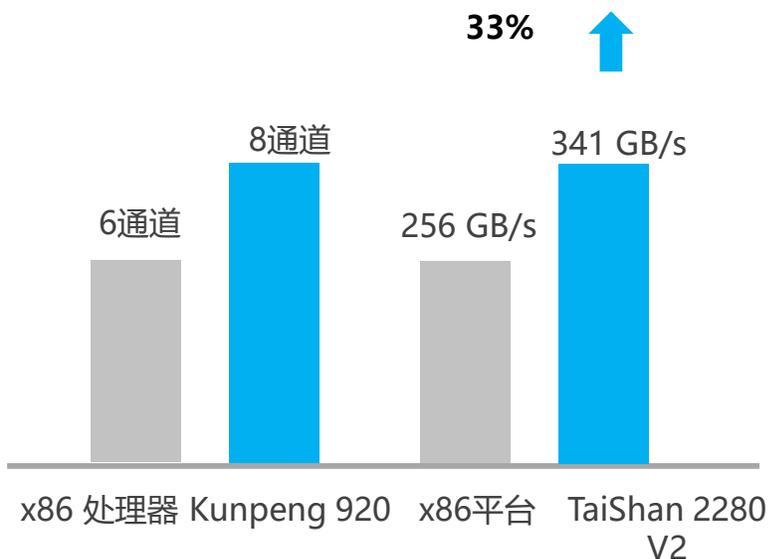
PCIe
4.0

100GE

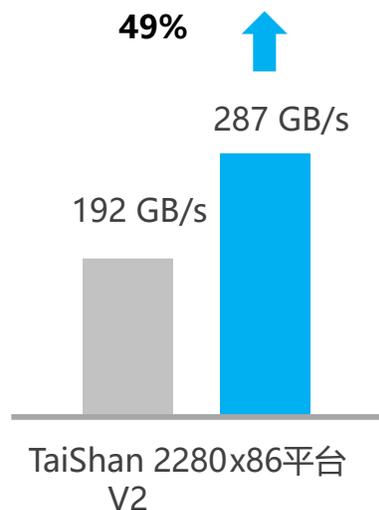


8通道内存技术提供高内存带宽

Kunpeng 8通道内存技术提升系统内存带宽



Stream内存带宽测试结果



平台	TaiShan 200	x86 server
处理器	2*Kunpeng 920-4826	2*x86 Skylake 6148
内存	16*DDR4-2666	12*DDR4-2666
Stream内存带宽	287 GB/S	192 GB/S
效率	84.28%	75.08%

*以上测试数据或功能实现的结果来源于华为实验室中TaiShan服务器测试结果。其他测试条件下，结果可能存在偏差。

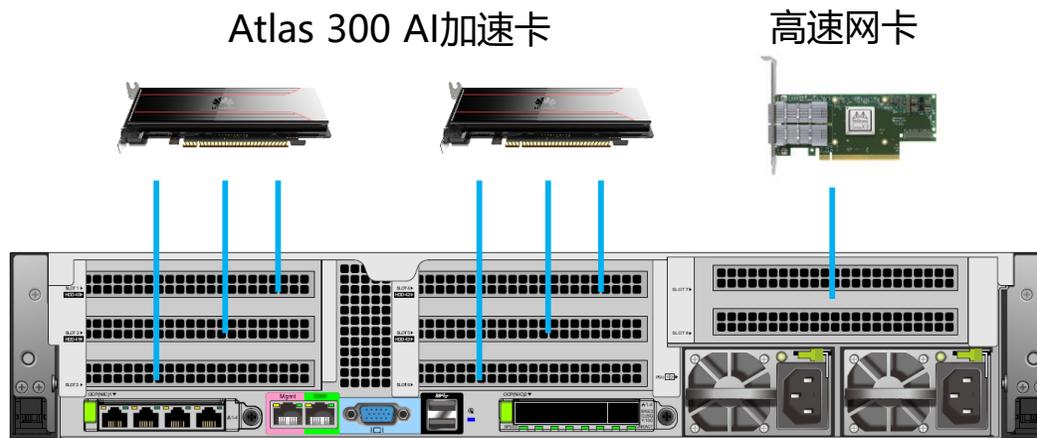
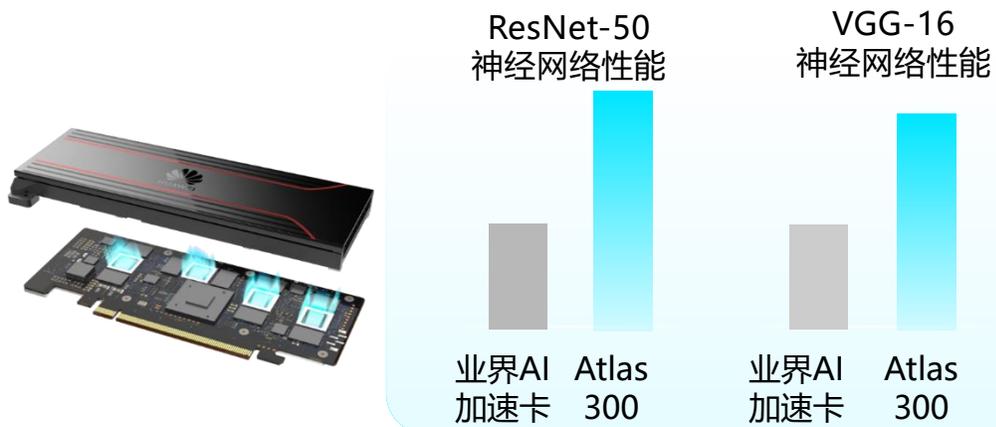


基于Atlas 300 AI加速卡的强大算力

基于昇腾AI芯片的Atlas 300 AI加速卡

可配置6块Atlas 300 AI加速卡+2块高速网卡

实测性能业界领先



视频分析 | OCR | 语音识别 |
精准营销 | 医疗影像分析



支持自研灵活插卡和智能网卡，实现丰富的网络配置

网络板载直出，灵活插卡形态



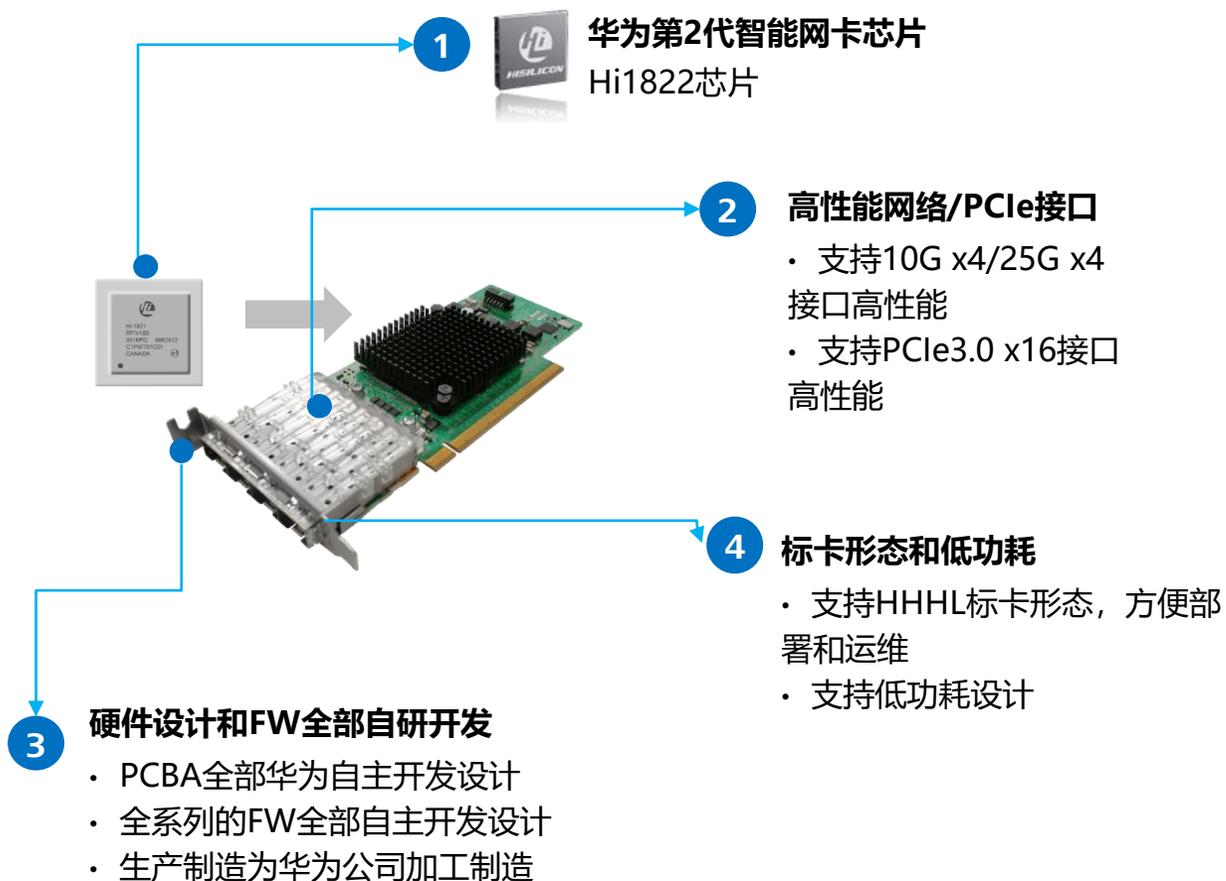
- 4*GE灵活插卡



- 4*10GE或者4*25GE灵活插卡



IN200智能以太网卡，标准网卡形态





支持华为ES3000 V5 NVMe SSD

1 高性能PCIe/SAS接口

- 聚焦NVMe PCIe3.0接口高性能
- 聚焦SAS 3.0接口高性能

2 自研SSD控制器芯片

- 华为第4代自研控制器

3 高品质NAND Flash颗粒

- 采用业界主流大厂高品质Flash颗粒

4 ES3000系列品牌

- ES3000系列为华为服务器品牌，聚焦NVMe/SAS高性能规格
- 1. ES3000P V5: 2.5' PCIe NVMe SSD
- 2. ES3000S V5: 2.5' SAS SSD
- 3. ES3000C V5: HHHL AIC SSD

5 硬件设计和FW全部自研开发

- PCBA全部华为自主开发设计
- 全系列的FW全部自主开发设计
- 生产制造为华为公司加工制造

更快: Faster, and Faster

- 采用新一代自研智能 SSD控制器Hi1812E
- 读写带宽及IOPS性能大幅提升
- 支持NVMe/SAS双模

更智能: Smarter

- 支持智能多流技术, 优化业务IO, 性能提升15%
- 支持智能监控管理、智能维护, 运维更高效

更简单: Simpler

- 面向客户应用进行优化, 提供参考架构,
- 提供基于应用场景的优化配置

ES3500P V5
Solid State Drive
ES3600P V5
Solid State Drive



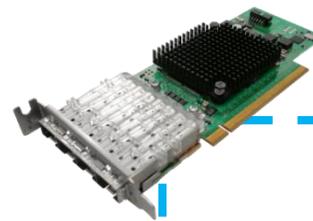
TaiShan服务器：计算芯片全自研，安全可靠

昇腾AI芯片



华为Atlas 300 AI加速卡

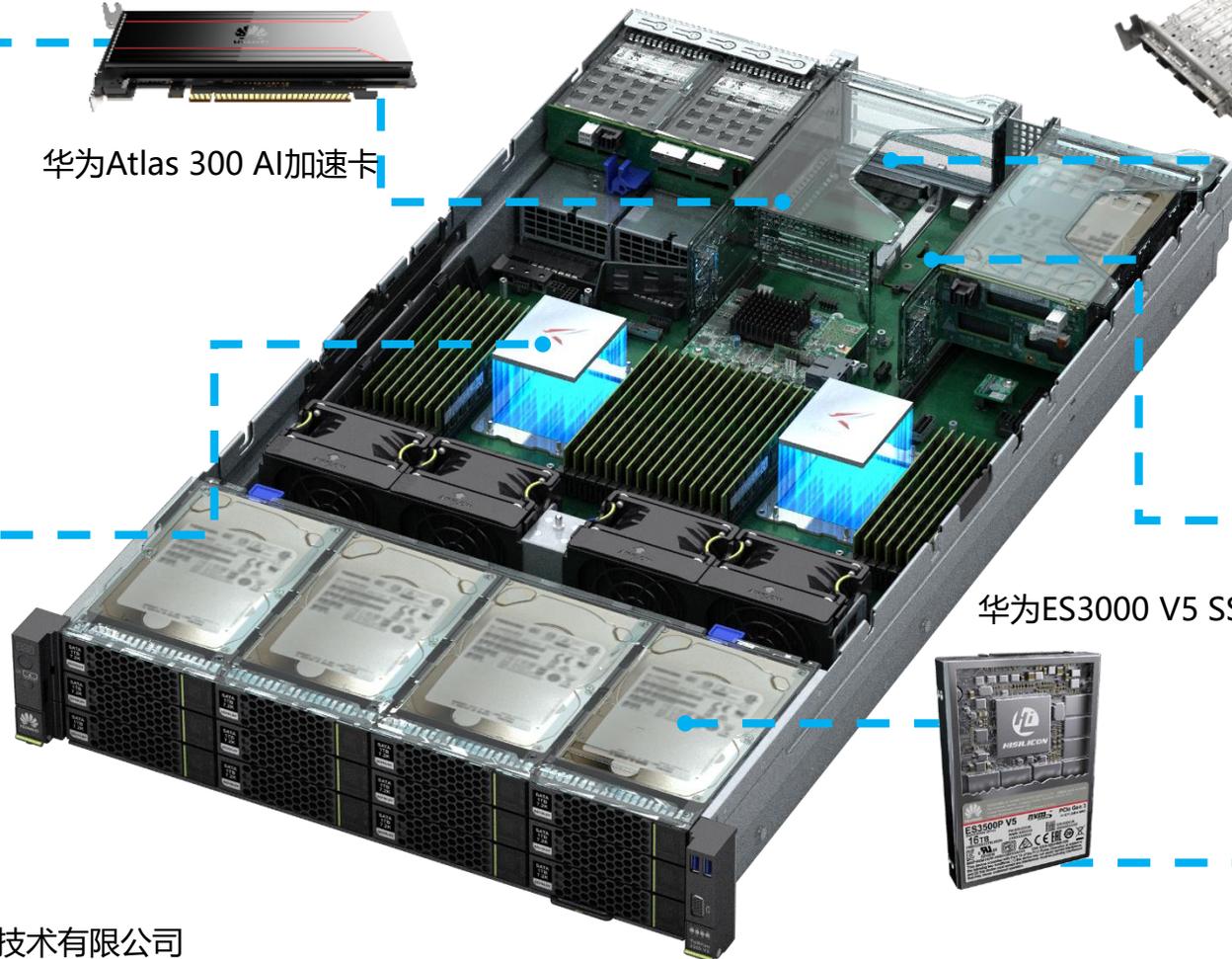
智能网卡芯片



华为“IN”智能网卡



Kunpeng 处理器



智能管理芯片



华为ES3000 V5 SSD

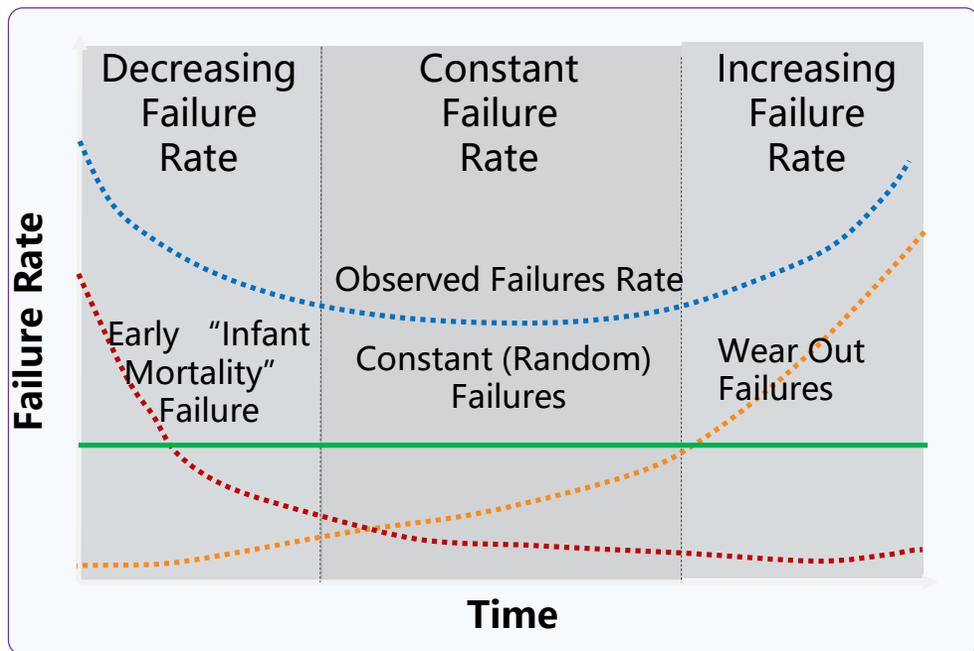


智能SSD控制器芯片



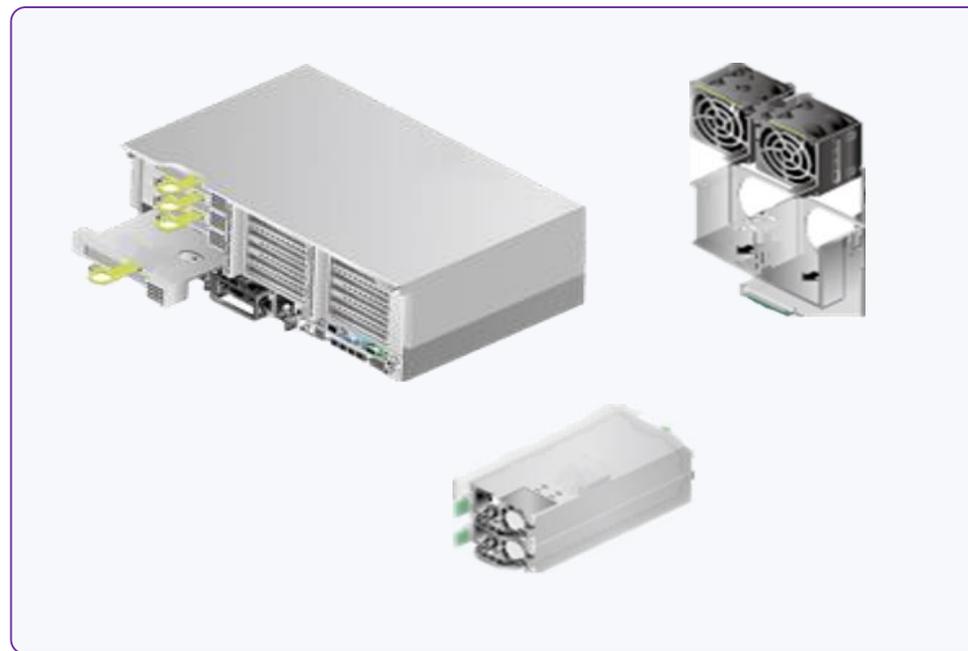


质量从源头做起：器件降额选件与冗余设计



降额选件

- 电信级的元器件供应商认证流程
- 降额设计，设计额定值远高于承受应力



冗余热插拔

- PSU/FAN等易损部件冗余设计
- PCIe部件支持热插拔(部分机型)



高可靠设计：服务器三重抗振提升质量

- 机械硬盘故障率在服务器中占比高。
- 通过硬盘三重防振设计，提升稳定性，降低硬盘故障率。

6.25%

服务器硬盘故障率



减振螺钉

振动源头：8个减振螺钉，减少风扇振动对风扇框的影响



刚度加强

振动路径：机框刚度加强，缓解振动传导



弹片&阻尼垫片

受体保护：硬盘托架上新增金属弹片与阻尼垫片

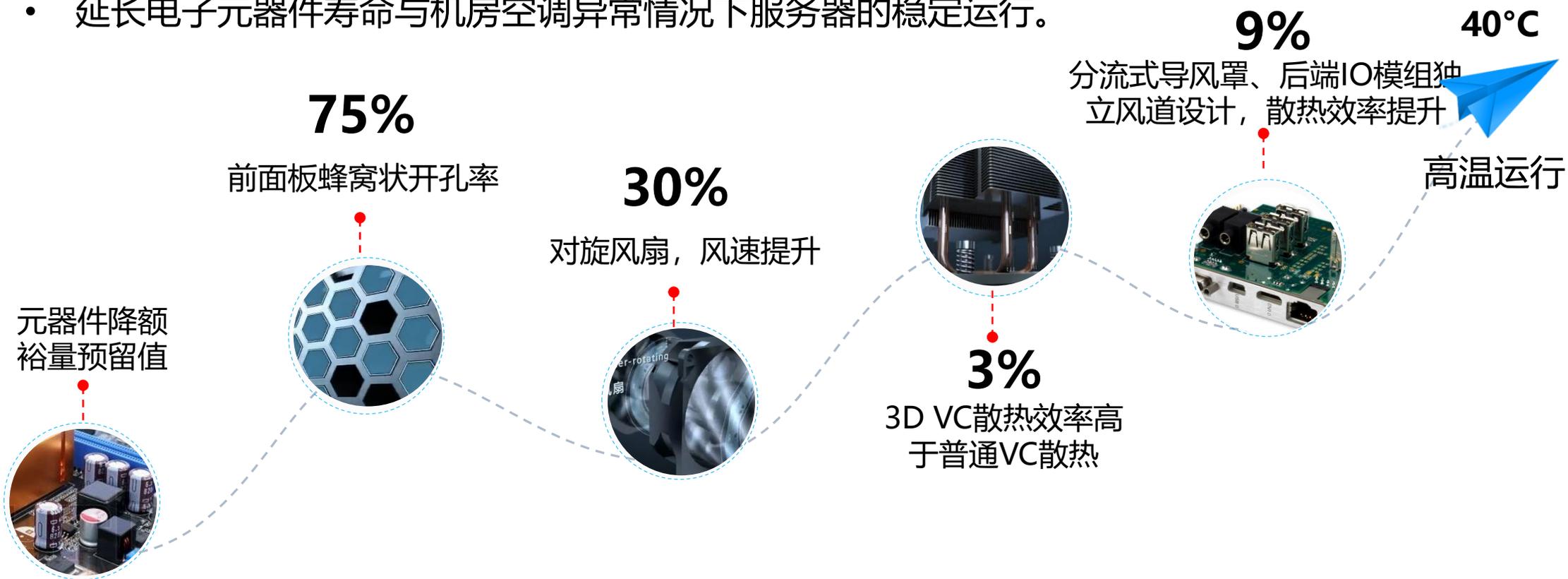
2.5%

华为服务器硬盘故障率



高可靠设计：高效散热支持更高环温

- 从元器件选型，到沿风向运行路径，端到端散热设计提升散热效率。
- 延长电子元器件寿命与机房空调异常情况下服务器的稳定运行。





目录

1. 鲲鹏生态介绍
2. 华为鲲鹏处理器介绍
- 3. Taishan服务器介绍**
 - TaiShan服务器概述
 - TaiShan 200机架服务器产品介绍
 - TaiShan 200机架服务器价值特性
 - TaiShan 200高密服务器介绍
4. 华为云鲲鹏云服务介绍
5. 鲲鹏操作系统介绍



X6000超级机框规格

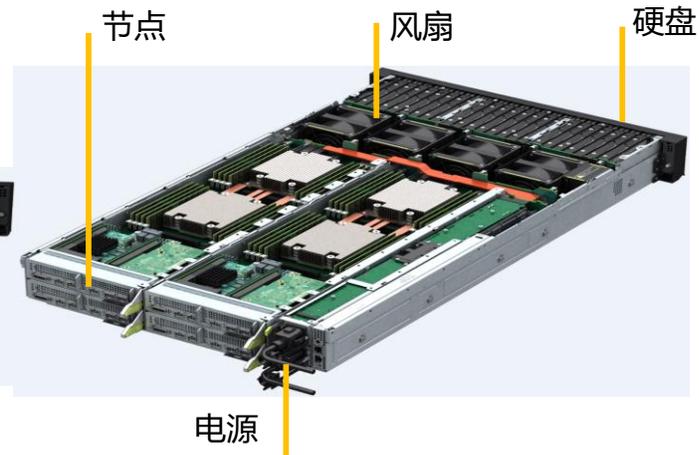
前视图



后视图



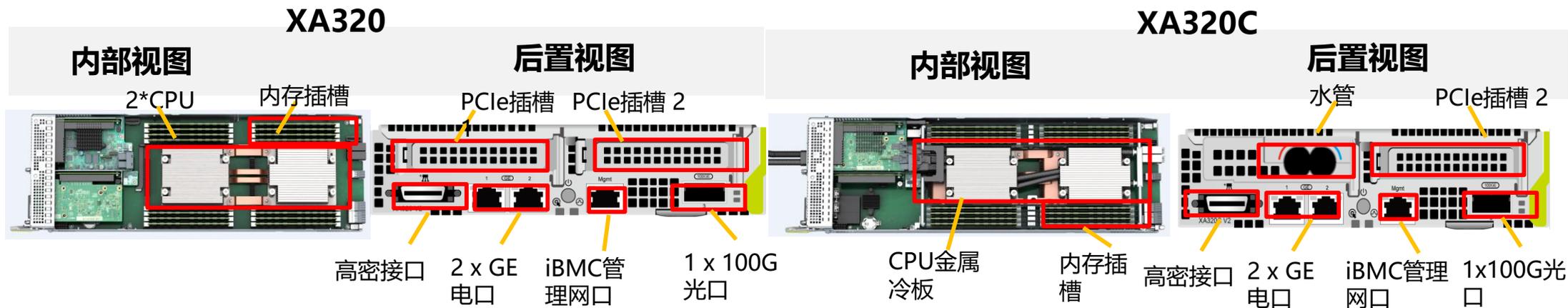
内部视图



产品	高密服务器
形态	2U机框, 支持4个XA320/XA320C半宽服务器
电源	2个热插拔3000W电源模块, 支持1+1冗余
供电	支持100~240V AC, 240V DC
风扇	4个热拔插风扇模组, 支持N+1冗余
工作温度	5°C- 35°C
尺寸(宽x深x高)	436mm x 818.9mm x 86.1mm



XA320&XA320C计算节点规格



形态	XA320计算节点	XA320C计算节点	说明
处理器	2*Kunpeng920 (48核2.6G)	2*Kunpeng920 (48核2.6G/64核2.6G)	64核2.6G风冷Q3支持
内存插槽	16个DDR4 2933		目前最大支持32GB, 64GB在Q3支持
硬盘数量	支持2~6个2.5" SAS/SATA硬盘		1、NVMe在Q3支持 2、风冷目前只支持2个硬盘, Q3优化后48核可以支持6个硬盘 3、风冷64核只能支持2个硬盘
板载网络	2*GE电口+1*100GE光口		
PCIe扩展	支持2个PCIe半高半长的标准扩展插槽	支持1个PCIe半高半长的标准扩展插槽	4*25G标卡 (1822) 在Q3支持
工作温度	5°C- 35°C		35°C环境温度时每节点最大支持2个硬盘
尺寸 (宽x深x高)	177.9mm x 545.5mm x 40.5mm		



支持3000W电源和液冷散热，释放HPC计算潜力

和英特尔可扩展系列处理器算力对比 (2P)

CPU型号	总核数	主频 GHz	最大TDP	内存带宽	CFD应用性能*
2*黄金级6248	40	2.5	150	192GB/s	1
2*Kunpeng920-4826	96	2.6	150	287GB/s	1.2

X6000超级机框提升单机柜HPC算力 20%

- 相比x86 6248方案，CFD场景单机柜算力提升20%
- 相同空间下可提升用户业务性能20%
- 可缩减用户CFD仿真时间20%，加速业务研发

固定算力高密部署场景

方案	单柜机框数	机柜数	CFD应用性能*	节省空间
2*Kunpeng920-4826@风冷	15	4	240	33%
2*Kunpeng920-4826@全液冷	20	3	240	

X6000超级机框节省HPC集群机柜数量 33%

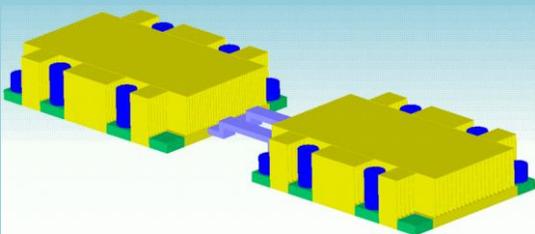
- 构建固定应用算力HPC集群，X6000风冷散热需要部署4个机柜
- X6000全液冷散热仅需要3个机柜

*使用openFOAM评估，测试数据为华为实验室数据，不同环境可能存在误差。



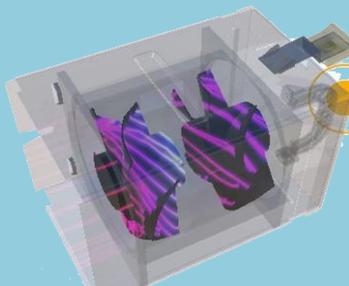
独特系统散热设计

连体散热器



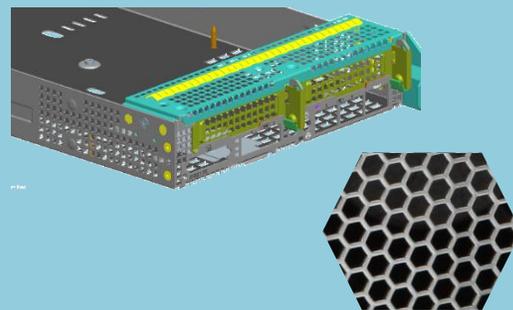
前后CPU温差控制在 3°C 内，确保CPU处于稳定工作状态

高风压对旋风扇



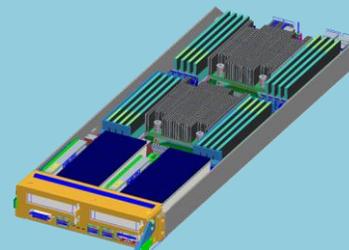
提供强大的系统冷却用风量，风量提升10%

整机开孔设计



优化背板、面板和机框开孔设计，提升机框通风能力，开孔率达到66%

全覆盖温度传感器



覆盖关键器件和风道位置，实现精确智能控温



思考题

1. TaiShan 200机架服务器包含哪些型号?

A.2280

B.5280

C.2480

D.X6000



本节小结

- 本节主要介绍了TaiShan 200服务器的产品，包括机架服务器和高密服务器，以及其技术特点。通过本节的学习，应掌握TaiShan 200服务器的型号、特点及应用场景等知识点。



目录

1. 鲲鹏生态介绍
2. 华为鲲鹏处理器介绍
3. Taishan服务器介绍
- 4. 华为云鲲鹏云服务介绍**
 - 华为鲲鹏云服务背景
 - 华为鲲鹏云服务重点内容
5. 鲲鹏操作系统介绍



本节概述和学习目标

- 本节内容介绍的是华为云鲲鹏云服务。
- 学完本课程后，您将能够掌握：
 - 华为云鲲鹏云服务的全景
 - 常用云服务的使用场景



企业数字化转型面临的挑战

管理/运维能力分散，水平不高



管理效率/水平难以跟上业务的发展

业务孤岛，投入分散，不能形成合力

初期投入小
Capex → Opex



创新业务试错，业务的不确定性，初期无法大规模投入

大规模投入交付周期长，无法满足业务快速变化的需要

业务扩展
应对业务变化慢



业务浪涌式或爆发式增长，企业难以快速响应

上线周期长，难以把握机会窗

业务可靠性
数据安全薄弱



企业业务和数据可靠性安全性受限于技术、资金、人才等因素，存在隐患和不到位



云服务是企业数字化转型的趋势

设备资源需求	统一通信	应用需求	服务需求	管理和安全
计算资源 存储资源 办公外设	多媒体会议 即时消息 电话 办公集成	企业邮件 企业网站 综合办公 CRM ERP	业务咨询 系统规划 电子商务 运作营销	管理维护 信息安全 访问安全 接入安全



提升客户服务

- 依托云服务的弹性拓展能力，增强业务服务能力，确保业务体验
- 利用新技术服务如AI、IoT，进行业务创新

降低运作成本

- 降低生产系统运作成本
- 降低通信和IT系统的总体拥有成本

提高业务处理效率

- 员工高效率业务处理及协作
- 员工外出时的远程和移动办公



华为云鲲鹏云服务概述

华为云鲲鹏云服务基于鲲鹏处理器等多元基础设施, 涵盖裸机, 虚机, 容器等形态, 具备多核高并发特点, 非常适合AI、大数据、HPC、云手机/云游戏等场景。

产业发展



政府



金融



大企业



互联网

...

全行业多场景覆盖

解决方案

全栈
专属云

HPC

大数据

企业应用

原生应用

AI

全栈鲲鹏
云服务



鲲鹏BMS



鲲鹏ECS



鲲鹏EVS



鲲鹏OBS



鲲鹏SFS



鲲鹏CCE



鲲鹏CCI



鲲鹏VPC



鲲鹏ELB



鲲鹏NAT网关



鲲鹏全系列云服务



华为鲲鹏
920
高性能CPU



Hi181x
存储控
制器



Hi182x
网络控
制器



Hi171x
服务器
管理



昇腾
310/910
高性能AI

核心基础设施



华为云鲲鹏云服务关键技术长周期投入

12年+云计算研发经验



华为云鲲鹏云服务
HUAWEI CLOUD

长期技术积累:

芯片、云



华为云鲲鹏云服务列表

IaaS		Virtual Private Cloud	Intelligence Anti-Fraud	PaaS	
Elastic Cloud Server	Elastic Load Balance	Database Security Service	ServiceStage	Application Operations Management	
Bare Metal Server	Elastic IP	Data Encryption Workshop	Distributed Message Service for Kafka	Distributed Cache Service for Redis	
Cloud Container Engine	VPC Endpoint	Intelligent CAPTCHA Service	Application Performance Management	OceanConnect Device Management	
Cloud Container Instance	NAT Gateway	Situation Awareness	OceanConnect Device Access	OceanConnect Global SIM Link	
Cloud Phone	Cloud Eye	Cloud Bastion Host	OceanConnect Edge	AI	
Dedicated Computing Cluster	Simple Message Notification	SSL Certificate Manager	MapReduce Service	Cloud Search Service	
Dedicated Host	Domain Name Service	CDN		Data Integration Service	
Elastic Volume Service	Domain Registration	Content Delivery Network	DataBase		Data Lake Insight
Dedicated Distributed Storage Service	Auto Scaling	RDS for MySQL	RDS for PostgreSQL		Data Lake Visualization
Storage Disaster Recovery Service	Cloud Connect	Document Database Service	IoT		Data Warehouse Service
Dedicated Enterprise Storage Service	Virtual Private Network	OceanConnect IoV		DAYU	
Cloud Server Backup Service	Security		OceanConnect Smart Campus		
Data Express Service	Anti-DDoS	OceanConnect Studio	OceanConnect Stream Orchestration		
Software Repository for Container	Advanced Anti-DDoS	OceanConnect Provisioning			
Direct Connect	Host Security Service				
Image Management Service	Container Guard Service				
Scalable File Service	Web Application Firewall				
Object Storage Service	Vulnerability Scan Service				
Volume Backup Service					



目录

1. 鲲鹏生态介绍
2. 华为鲲鹏处理器介绍
3. Taishan服务器介绍
- 4. 华为云鲲鹏云服务介绍**
 - 华为鲲鹏云服务背景
 - 华为鲲鹏云服务重点内容
5. 鲲鹏操作系统介绍



华为鲲鹏弹性云服务器概述

- 鲲鹏弹性云服务器（ECS）。基础云服务之一，也是用户可以直接感知到鲲鹏的最重要的服务。用户可以用过ECS直接购买鲲鹏云服务器，为云服务器添加磁盘、网络等资源，使其成为开发环境或者生产业务集群的一部分。



华为鲲鹏弹性云服务器型号和场景

主要型号

通用计算型KC1
CPU:MEM = 1:2/1:4
最高30G内网带宽, 400万
PPS

超高IO型KI1
最大60U480G
8 x 3.2TB

内存优化KM1
CPU:MEM = 1:8
最大60U480G

华为鲲鹏计算加速
Ascend 310/910



对应场景

建站、电商、游戏、
视频、大数据、HPC等

分布式缓存数据库、
大数据

Redis、Cassandra
内存数据库

AI训练、推理



华为鲲鹏弹性云服务器优势



全场景覆盖

- 通用/存储/内存/计算加速型
- 最大规格60U480G



超高性价比

- 多核整型算力领先
- 内存带宽领先
- 综合性价比领先



超高稳定可靠

- 全栈灾备保护
- 集成HSS服务, 抗攻击能力强
- 秒级故障恢复、多维故障隔离
- 千级虚拟机, 分钟级发放
- 秒级细粒度监控



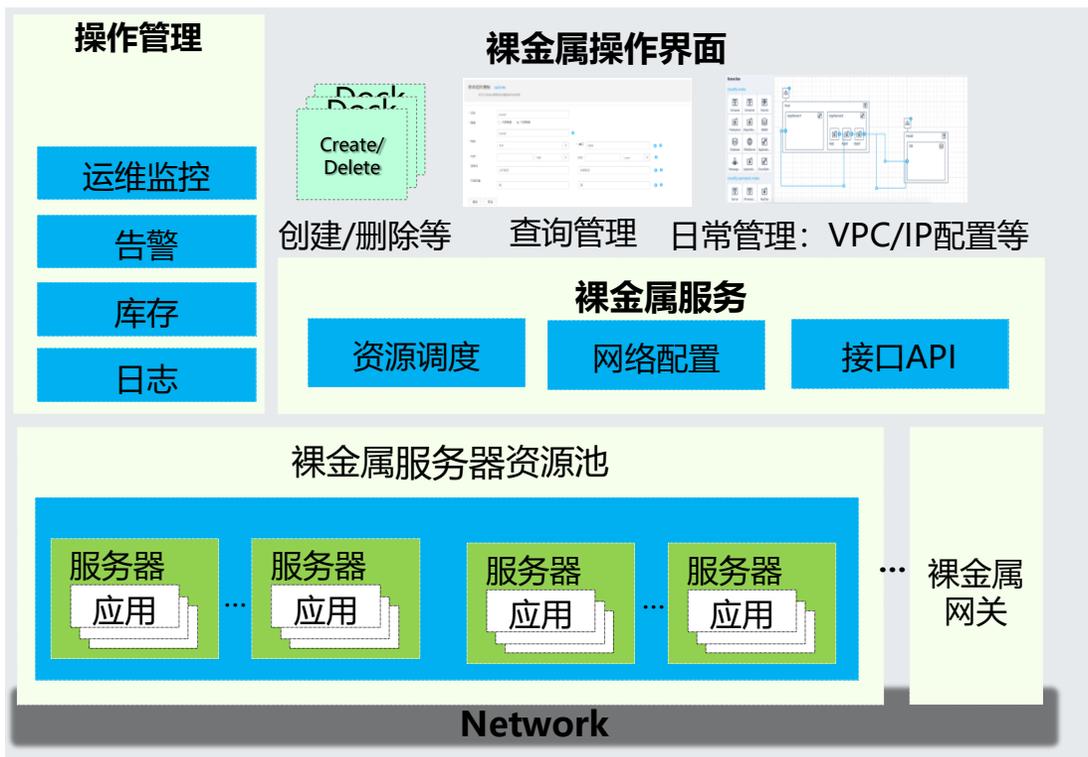
丰富生态

- 已兼容20+款主流操作系统
- 100+企业业务应用
- 凌云计划, 持续打造鲲鹏生态



华为鲲鹏裸金属服务器(BMS)概述

为云用户提供专属的物理服务器，提供卓越的计算性能，满足核心应用场景对高性能及稳定性的需求，结合了传统托管服务器带来的稳定性能与云中资源高度弹性的优势。



服务特点

- **易使用**：自助申请，一键式console操作，最快5分钟发放
- **云硬盘**：共享存储按需申请，支持共享卷，支持集群应用
- **灵活网络配置**：对接VPC、安全组，实现ECS与裸机、裸机与裸机之间通信
- **高可靠性**：数据自动备份，设备自动迁移
- **易维护**：可监报告警、硬件状态，提供远端console以及裸机带内监控
- **灵活付费**：包年、包月付费；非业务占用时间仅仅是裸机发放时间，时间短



华为鲲鹏裸金属服务器规格和场景

主要型号

通用型KS1

128Core、256G内存、
2*25G

存储型KD1

128Core、512G内存、
2*25G、12*10T

对应场景

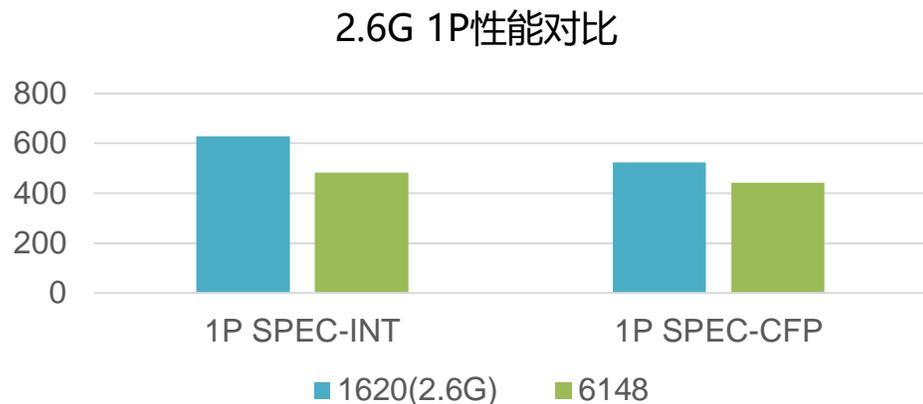
大数据管理节点、
NameNode、HPC等

大数据数据节点、
DataNode、Kafka等

众核算力领先同x86平台，适合高并发数据处理，如大数据、HPC、ARM原生应用场景



华为鲲鹏裸金属服务器优势



- 2.6Ghz单P性能为x86 Gold 6148（打开turbo）的**1.3**倍；
- 华为鲲鹏BMS相对x86相同配置（6148 CPU）在性价比上有优势；



高并发，性能领先

核数最高支持**128core**
全新SDI 3.0技术带宽提升**40Gbps**



分钟级发放

支持存储共享卷
在线获取、按需使用



智能加速

搭载自研AI芯片**Ascend 310/910**



华为鲲鹏计算服务总览

华为鲲鹏弹性云服务器				华为鲲鹏裸金属服务器	
通用计算型KC1	存储密集型KI1	内存密集型KM1	计算加速型 KAi1/KAt1	通用型KS1	存储型KD1
CPU: MEM=1:2/1:4 最高30G内网带宽 400万PPS	最大60U480G 8 x 3.2TB	CPU:MEM = 1:8 最大60U480G	Ascend 310/910	128Core、256G 内存、2*25G	128Core、512G内存、 2*25G、12*10T
建站、电商、游戏、视频、 大数据、HPC等	分布式缓存数据库、 大数据	Redis、Cassandra 内存数据库	AI训练、推理	大数据管理节点、 NameNode、 HPC等	大数据数据节点、 DataNode、Kafka等
2019.9.16商用	2019.10公测	2019.9.16商用	KAi1 2019.10月公测 KAt1 2020Q2公测	2019.9.16商用	2019.9.16商用



华为鲲鹏存储服务：基于自研芯片的全场景存储实例

华为云存储服务，全套自研芯片版本发布半年，销售容量>1EB+

面向自研的全闪存设计



- **业界最快SSD**：控制芯片，内置核心FTL算法
- **50%性能**：NVME架构，CPU直接与SSD通话

基于自研芯片的可靠性增强

- **可变EC效率提升15%**：鲲鹏多核算力增强可变EC效率，相较三副本性能不下降，实现免降级写，无回拷等可靠性增强
- **业务无感知的故障切换**：通过自研智能网卡1822，实现节点故障切换时间<1S（普通网卡>10S）

基于鲲鹏的三大算法优化

- **时延优化20%**：基于鲲鹏多核算力优势，优化CPU分组分核算法
- **时延波动率<5%**：基于鲲鹏内存带宽优势，优化Cache算法
- **IO并发和带宽提升100%**：鲲鹏多核算力优化IO调度算法，数据I/O第一优先级

关键芯片100%自研打造



- **鲲鹏920**，高性能CPU
- **Hi181x**，存储控制器
- **Hi182x**，网络控制器
- **Hi171x**，服务器管理



华为鲲鹏网络服务：全系列网络服务提供

新一代云网络平台：统一资源池及管理、网络容量横向扩展、租户隔离,故障隔离

全自研硬件

- Kunpeng 920
- Hi182X 网络控制器
- 交换机CE168XX
- SDI卡，管理/转发卸载



鲲鹏系列网络性能

1. 100G ROCE实现RDMA网络，时延15 μ s，业界领先，支持HPC、EI大数据场景对更快网络，更小时延的诉求
2. HTTPS卸载，性能提升3倍

鲲鹏系列网络管控面

- 1.单Region 20-30万主机级的大规模网络，提升100%
- 2.线下通用服务器、裸机等互联互通;
- 3.可视化、智能化运维



华为云鲲鹏容器概述

- 容器(container): 容器是一种虚拟化实例，一个操作系统的内核允许多个隔离的用户空间实例。不像虚拟机（VM），容器不需要为每个实例运行一个完备的操作系统（OS）镜像。相反，容器能够在单个共享操作系统里面运行应用程序的不同实例。
- 华为云提供高性能、高可用、高安全的企业级容器服务，通过CNCFO官方认证的两种Kubernetes服务供用户选择，包括云容器引擎（CCE）与云容器实例（CCI）。



华为云鲲鹏容器服务型号和场景

主要型号

华为鲲鹏
Kubernetes容
器 (CCE)

华为鲲鹏
Serverless容
器 (CCI)



对应场景

建站、电商、视频、大
数据、混合计算，成本
优化等

基因、大数据、
Serverless、应用AI训
练、推理



华为鲲鹏CCE

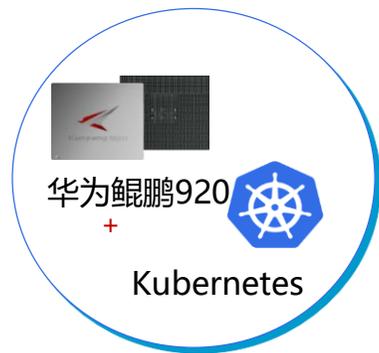
云容器引擎（Cloud Container Engine）提供高可靠高性能的企业级容器应用管理服务，支持Kubernetes社区原生应用和工具，简化云上自动化容器运行环境搭建。



游戏、直播、视频、金融、汽车、航空、能源

华为鲲鹏Kubernetes容器

多平台混部 跨云管理 一键式交付



多平台混部

支持x86与华为鲲鹏节点同时部署到一个集群中，及应用跨平台节点的扩容



跨云管理

跨华为云与其它公有云、私有云的K8S资源和应用统一管理



一键式交付

一键完成多平台镜像构建和部署，降低应用跨平台迁移难度



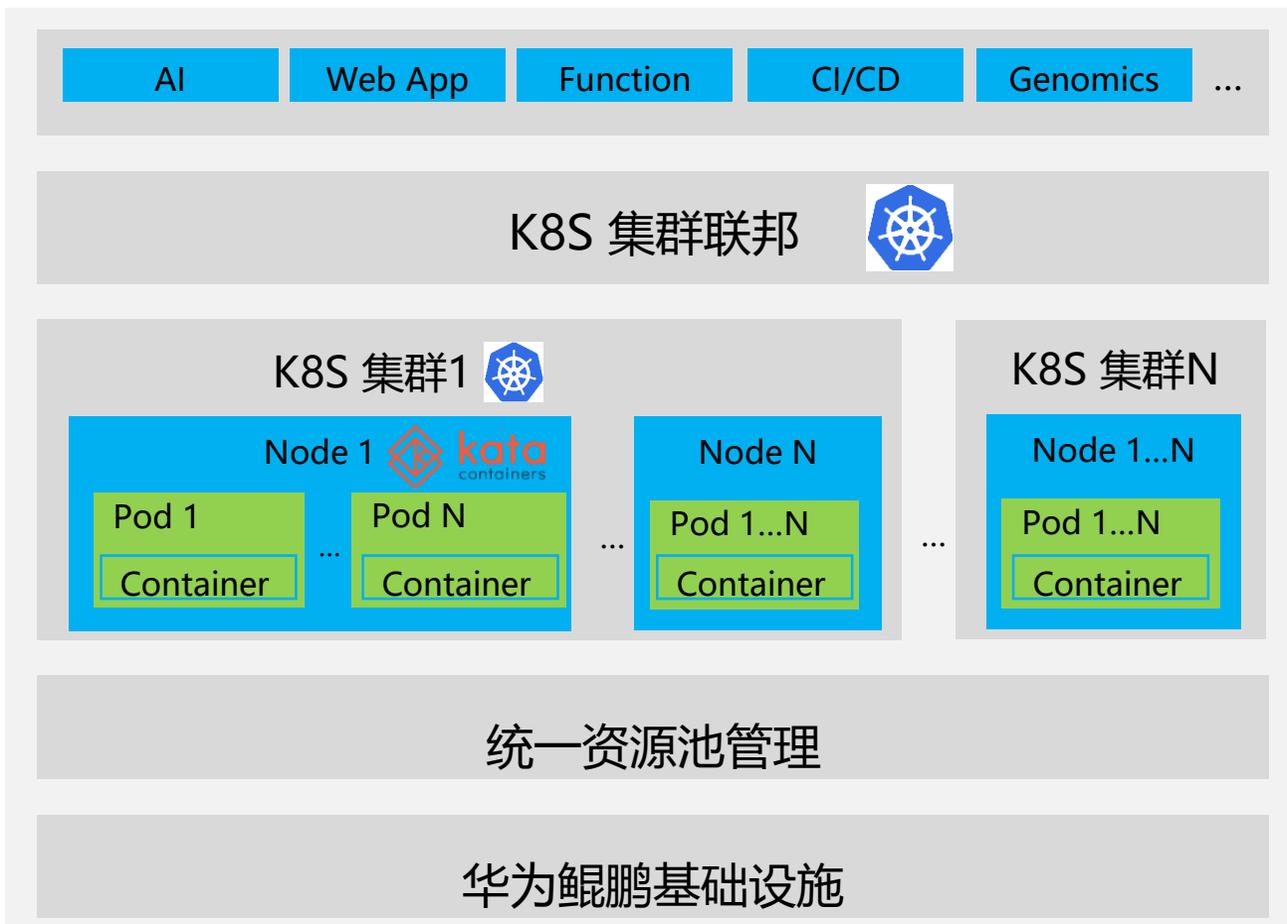
高性能

面向多核计算场景，华为鲲鹏多核优势明显，有效提升计算性能



华为鲲鹏CCI

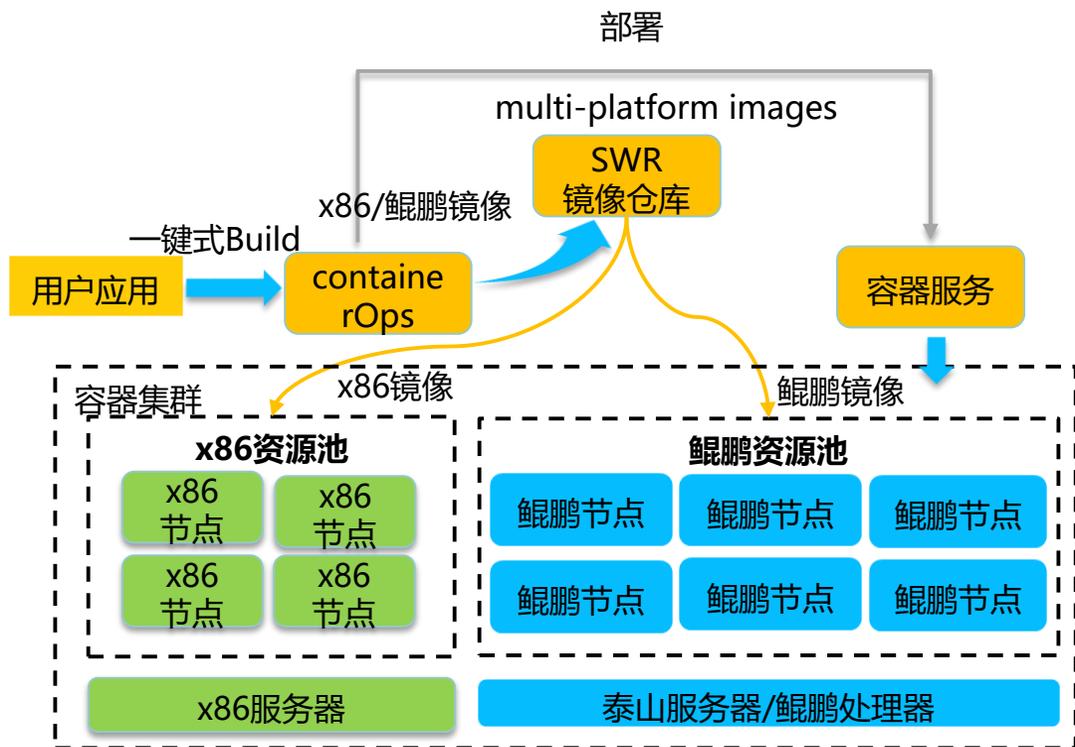
云容器实例（Cloud Container Instance，CCI）服务提供 Serverless Container（无服务器容器）引擎，无需创建和管理服务器集群即可直接运行容器。



- **免运维**：用户无需感知集群和服务器，直接部署容器应用，大幅提升容器易用性；
- **高安全**：同时具备容器级别的启动速度和虚拟机级别的安全隔离能力，提供更好的容器体验；
- **极致性能**：支持异构计算加速、高性能网络和存储、优化的调度，满足高性能计算诉求；
- **秒级计费**：根据实际使用的资源数，按需按秒计费，避免业务不活跃时的费用开销，降低用户成本；
- **开放生态**：原生支持K8S、KATA和Docker接口，兼容开源社区生态，方便企业探索属于自己的云原生模式；
- **极速弹性**：提供秒级弹性伸缩能力，轻松应对业务快速变化，稳健保障用户业务SLA。



华为云鲲鹏容器特性



多平台自适应混部

支持在同一容器集群中对x86和鲲鹏节点进行混合管理；根据节点CPU架构，自动拉取相应平台的容器镜像，实现应用从x86到华为鲲鹏架构的自动适配、跨平台弹性扩容、分批次迁移

应用无感迁移

为用户提供一键式从源码到多平台容器镜像版本的构建、部署、发布能力，用户可以快速将原有的x86应用迁移到华为鲲鹏平台

高性能

速度优： 华为鲲鹏容器 NUMA自动亲和优化，发挥CPU计算最佳性能

发放快： CCI提供鲲鹏容器秒级发放，Volcano支持鲲鹏集群发放速度达1000容器/秒

转发高： 鲲鹏容器网络优化，支持最高30G内网带宽，服务网格转发性能提升50%

极速AI容器

提供华为鲲鹏+Ascend 310 细粒度推理容器

提供华为鲲鹏+Ascend 910 1024卡高性能容器分布式训练集群

Volcano支持MindSpore、Tensorflow等多种AI训练框架，提供训练任务算力灵活调度分配，支持AI任务的调度亲和优化，多任务并行资源优化。



华为云鲲鹏容器优势

华为鲲鹏容器高性能批量计算平台

AI模型训练与推理

性能

50%

海量数据分析

成本

30%

基因测序

效率

10倍

AI容器解决方案



大数据容器解决方案



基因容器解决方案



Volcano 容器高性能批量计算平台

鲲鹏Kubernetes容器

鲲鹏Serverless容器

x86



华为鲲鹏



昇腾



异构

x86



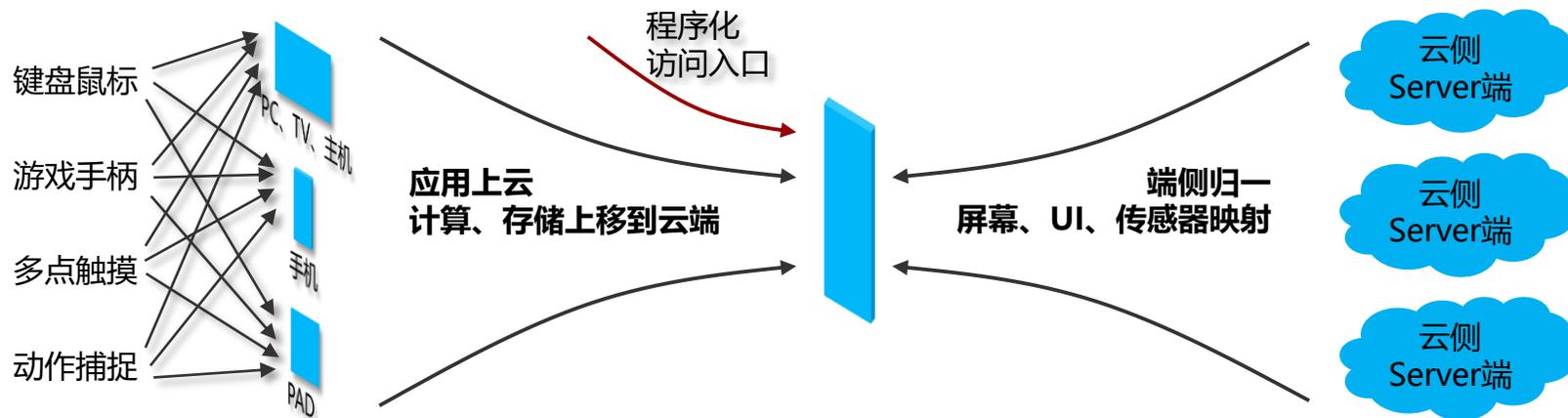
华为云鲲鹏容器服务总览

华为鲲鹏Kubernetes容器 (CCE)			华为鲲鹏Serverless容器 (CCI)	
通用华为鲲鹏容器集群	混合计算容器集群	AI鲲鹏容器集群	通用型鲲鹏容器实例	AI鲲鹏容器实例
支持华为鲲鹏弹性云服务器和 华为鲲鹏裸金属服务器	支持华为鲲鹏与x86服务器 混合集群	支持Ascend 310/910 集群	CPU:MEM=1:2~1:8灵活配置	Ascend 310推理实例 Ascend 910分布式训练实例
建站、电商、视频、大数据	混合计算, 成本优化	AI训练、推理	基因、大数据、Serverless应用	AI训练、推理
2019.9月公测	2019.10月公测	Ascend 310: 2019.12月公测 Ascend 910: 2020.Q2公测	2019.12月公测	Ascend 310: 2020.Q1公测 Ascend 910: 2020.Q2公测



华为云鲲鹏云手机概述

云手机：将终端侧上的游戏或应用运行在远端，而通过网络访问



7 × 24永久在线

无限续航
永不关机

规格灵活配置

规格灵活调整
性能无限拓展

云上升级换代

本地零占用
云端升级换代

企业级安全防护

数据安全
远程管理



华为云鲲鹏云手机场景

智能应用托管

海量部署，批量控制

性能&型号，灵活调整

安卓应用，无缝兼容

批量群控
移动应用云端智能运行

云游戏

密度提升10倍

时延低于100ms

8K游戏仅需8兆带宽

新型游戏管理平台
无需下载，即点即玩

移动安全办公

云端部署，数据不落地

集中管理和运维

弹性扩展，多机复用

公私数据分离，智能群控管理
降本增效，信息安全更有保障

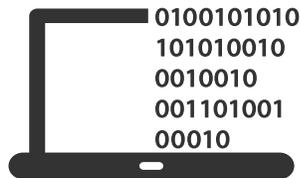


华为云鲲鹏云手机优势



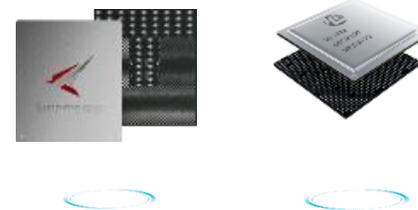
原生指令兼容性

32/64位应用全兼容
ARM原生指令兼容性



Monbox双OS共内核架构

单台服务器超200路并发
性能较模拟器提升10倍以上



自研芯片及硬件底座

华为鲲鹏916\920芯片
自研高性能GPU

全自研芯片加速，性能领先



思考题

1. 华为鲲鹏BMS云服务器最高可提供多少核?
 - A.32
 - B.48
 - C.64
 - D.128



本节小结

- 本节主要介绍了华为云鲲鹏云服务。内容包括云服务的背景，以及常用的华为云鲲鹏云服务，包括计算服务、存储服务、网络服务、容器服务以及云手机。
- 通过本节的学习，需要掌握常见华为云鲲鹏云服务的规格、特性及使用场景等内容。



目录

1. 鲲鹏生态介绍
2. 华为鲲鹏处理器介绍
3. Taishan服务器介绍
4. 华为云鲲鹏云服务介绍
5. **鲲鹏操作系统介绍**
 - 鲲鹏OS兼容性
 - openEuler介绍



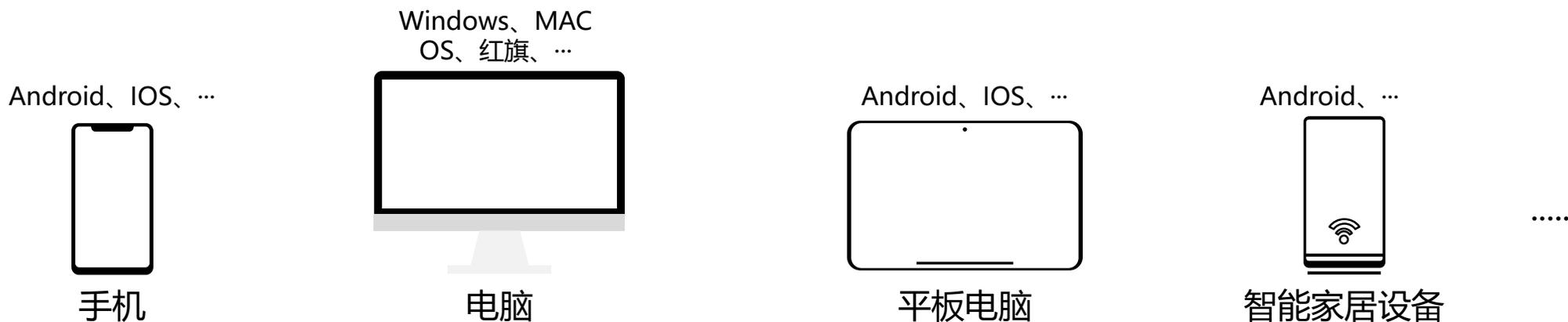
本节概述和学习目标

- 本章主要介绍鲲鹏兼容的操作系统以及openEuler的发展和特性等。
- 学完本课程后，您将能够：
 - 描述鲲鹏兼容的操作系统
 - 描述openEuler操作系统的特点以及openEuler开源社区的作用



操作系统介绍

- 操作系统是一管理计算机硬件与软件资源的程序，同时也是计算机系统的内核与基石。操作系统身负诸如管理与配置内存、决定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务。操作系统也提供一个让使用者与系统交互的操作接口。小到智能穿戴设备、电视盒子，大到电脑、服务器、路由交换设备，都有自己的操作系统。





Linux操作系统起源

- 1991年10月5日，Linus Torvalds在comp.os.minix新闻组织上正式对外宣布Linux内核的诞生，并且开源。
- 在开源社区的参与者努力下，1994年3月Linux 1.0内核版本正式发布。
- Linux是一套免费使用和自由传播的类Unix操作系统，是一个基于POSIX和Unix的多用户、多任务、支持多线程和多CPU的操作系统。





Linux操作系统的组成

- Linux操作系统构成
 - Linux内核
 - 应用程序
- Linux内核项目介绍
 - 主要作者：Linus Torvalds（芬兰）。
 - 1991年10月，发布Linux 0.02版（第一个公开版）。
 - 1994年3月，Linux 1.0版发布。
 - Linux内核的标志——企鹅Tux，取自芬兰的吉祥物。





常见的Linux操作系统



RedHat

CentOS

Fedora

Kali Linux

DVL

OpenSuse

Debian

Ubuntu

deepin

Oracle Linux

Arch Linux

中标麒麟

.....



华为鲲鹏处理器OS兼容性





CentOS简介

- 来自于Red Hat Enterprise Linux依照开放源代码规定释出的源代码所编译而成，CentOS不包含封闭源代码软件。CentOS是RHEL源码再编译的产物，而且在RHEL的基础上修正了不少已知的Bug，相对于其他Linux发行版，其稳定性值得信赖。
- CentOS在2014年加入红帽后，继续坚持提供免费操作系统的使用和更新。目前CentOS操作系统的使用领域非常广泛，适合服务器使用。





debian简介

- debian是迄今为止最遵循GNU规范的Linux系统。提供了接近十万种不同的开源软件支持，具有认可度和使用率，对于各类内核架构支持性良好，稳定性、安全性强，更有免费的技术支持。
- 基于其优秀的稳定性和资源开销小等特点，debian在虚拟专用服务器中应用比较多。





Ubuntu简介

- Ubuntu是Debian的一款衍生版，是一个以桌面应用为主的开源免费的操作系统。Ubuntu侧重于它在这个市场的应用，在服务器、云计算、甚至一些运行Ubuntu Linux的移动设备上很常见。
- Ubuntu具有庞大的社区力量，用户可以方便地从社区获得帮助。





deepin简介

- deepin操作系统是由武汉深之度科技有限公司开发的Linux发行版。deepin操作系统拥有自主设计的特色软件：深度软件中心、深度截图、深度音乐播放器和深度影音，全部使用自主的deepinUI。
- deepin操作系统专注于使用者对日常办公、学习、生活和娱乐的操作体验的极致，适合笔记本、桌面计算机和一体机。





中标麒麟简介

- 中标麒麟是中标Linux和银河麒麟合并后发布的一款Linux发行版本，中标麒麟操作系统采用强化的Linux内核，分成桌面版、通用版、高级版和安全版等，满足不同客户的要求，已经广泛的使用在能源、金融、交通、政府、央企等行业领域。



中标麒麟
NeoKylin



各行业常用的OS

行业	常用OS
电信运营商	CentOS等
银行、证券	CentOS、中标麒麟、银河麒麟等
互联网	企业定制OS、Debian等
政府、央企	CentOS、中标麒麟、银河麒麟、深度等
电力、能源	CentOS、中标麒麟、银河麒麟、Ubuntu等
交通运输	CentOS、中标麒麟等
安平	CentOS等
教育、医疗	CentOS、银河麒麟等



TaiShan服务器OS兼容性查询

- 《智能计算产品兼容性查询助手》网站链接如下：
 - <http://support.huawei.com/onlinetoolweb/ftca/index?serise=9>
- 在查询时，可以指定部件类型，这样查询到的OS兼容性信息更加精确。

操作系统兼容性 | 部件兼容性

* 产品型号: TaiShan 2280 V2

* 部件类型:

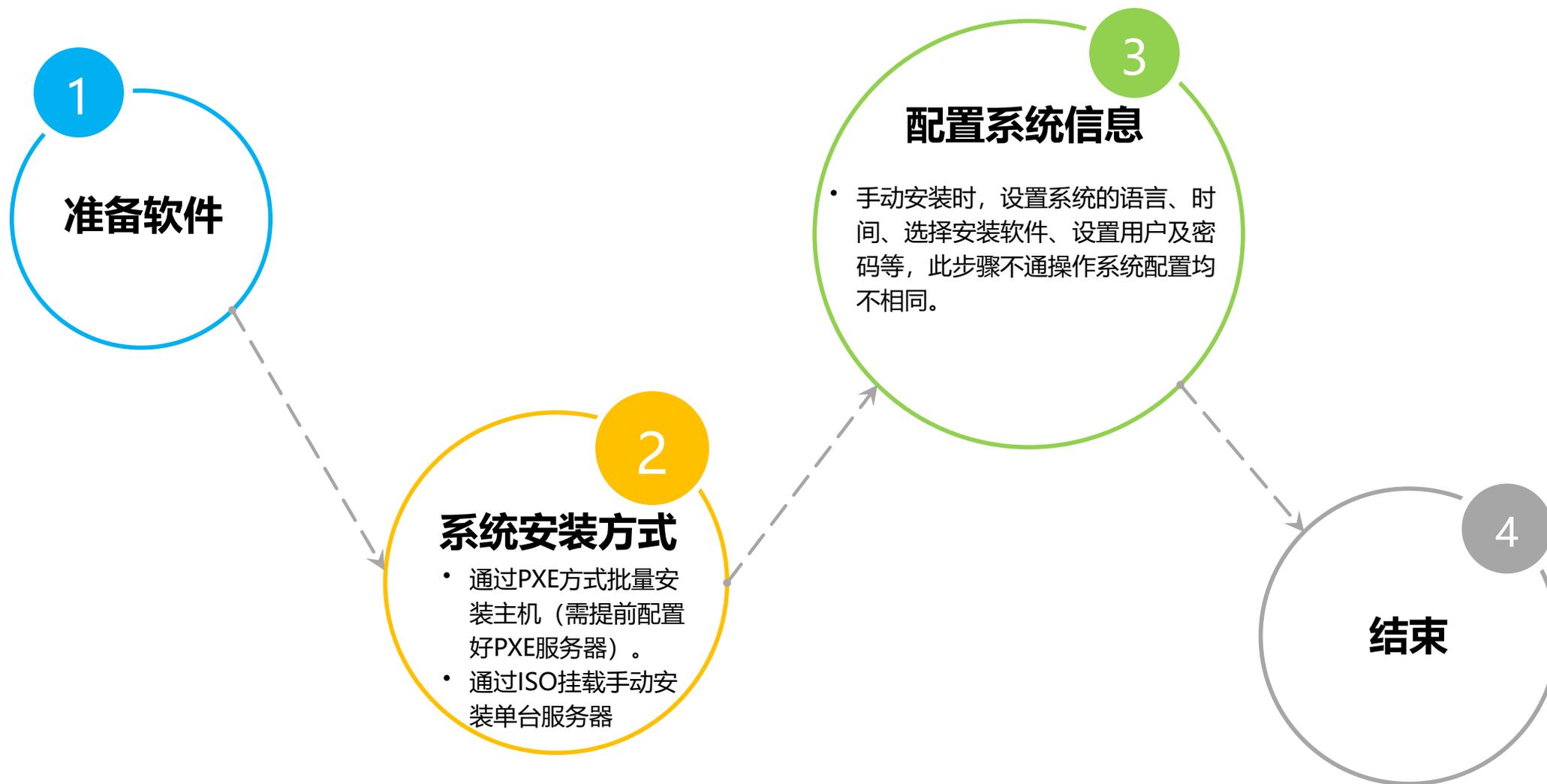
- CPU: [] []
- DIMM: [] []
- RAID控制卡: [] +
- Mezz卡: [] [] +
- PCIe卡: [] [] +

导出 Word | 导出 Excel | 导出 PDF | 重置 | 查询

操作系统名称	特性描述	认证链接	驱动下载链接	备注
BClinux 7.6 ARM	BClinux for ARM		Link	Note92;Note96;
CentOS 7.6 ARM	CentOS Linux 7 Update 6 Server for ARM 64		Link	Note87;Note92;Note93;
EulerOS V2.0 SP8 ARM	EulerOS V2.0 SP8 for ARM		Link	Note87;Note92;
NeoKylin Server V7.0 U6 ARM	NeoKylin Server V7.0 U6 ARM		Link	Note92;Note96;
SLES 15 SP1 ARM	SUSE Linux Enterprise Server 15 Service Pack 1 for ARM		Link	Note92;Note94;Note96;
Ubuntu 18.04.1 LTS ARM	Ubuntu 18.04.1 LTS Server Edition for ARM 64		Link	Note92;Note93;Note96;



操作系统安装流程





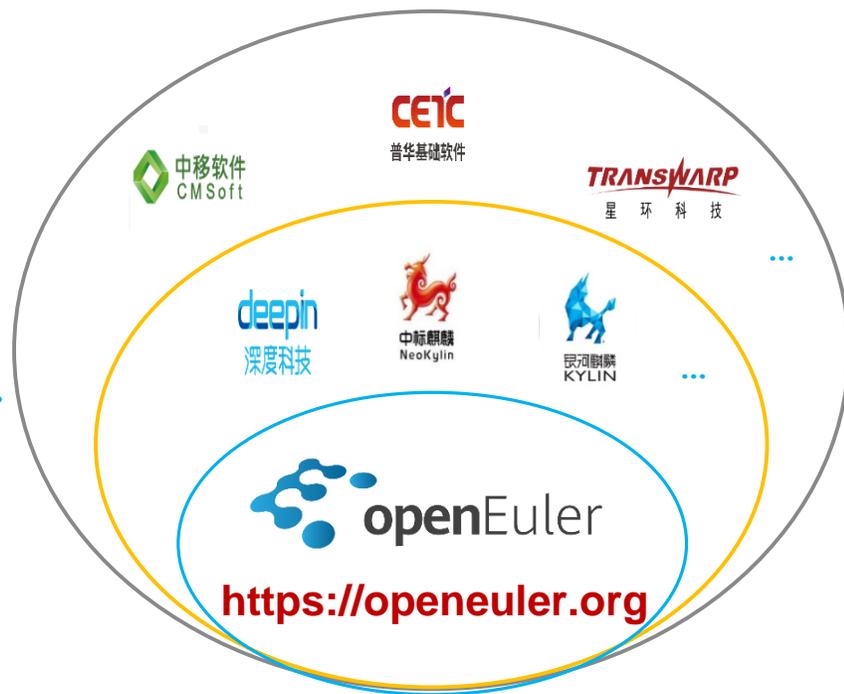
目录

1. 鲲鹏生态介绍
2. 华为鲲鹏处理器介绍
3. Taishan服务器介绍
4. 华为云鲲鹏云服务介绍
5. **鲲鹏操作系统介绍**
 - 鲲鹏OS兼容性
 - openEuler介绍



openEuler概述

国际开源社区



- openEuler有近10年的技术积累，已广泛用于华为内部产品配套。同时，华为基于对鲲鹏处理器的深刻理解，在性能、可靠性、安全性等方面对EulerOS进行了深度优化。为促进多样性计算产业发展及生态建设，华为把服务器领域的技术积累进行开源。
- openEuler 是一款开源操作系统。当前 openEuler内核源于Linux，支持鲲鹏及其它多种处理器，能够充分释放计算芯片的潜能，是由全球开源贡献者构建的高效、稳定、安全的开源操作系统。openEuler 适用于数据库、大数据、云计算、人工智能等应用场景。同时，openEuler也是一个面向全球的操作系统开源社区名称。

Sep 19, 2019

合作伙伴内测



deepin
深度科技

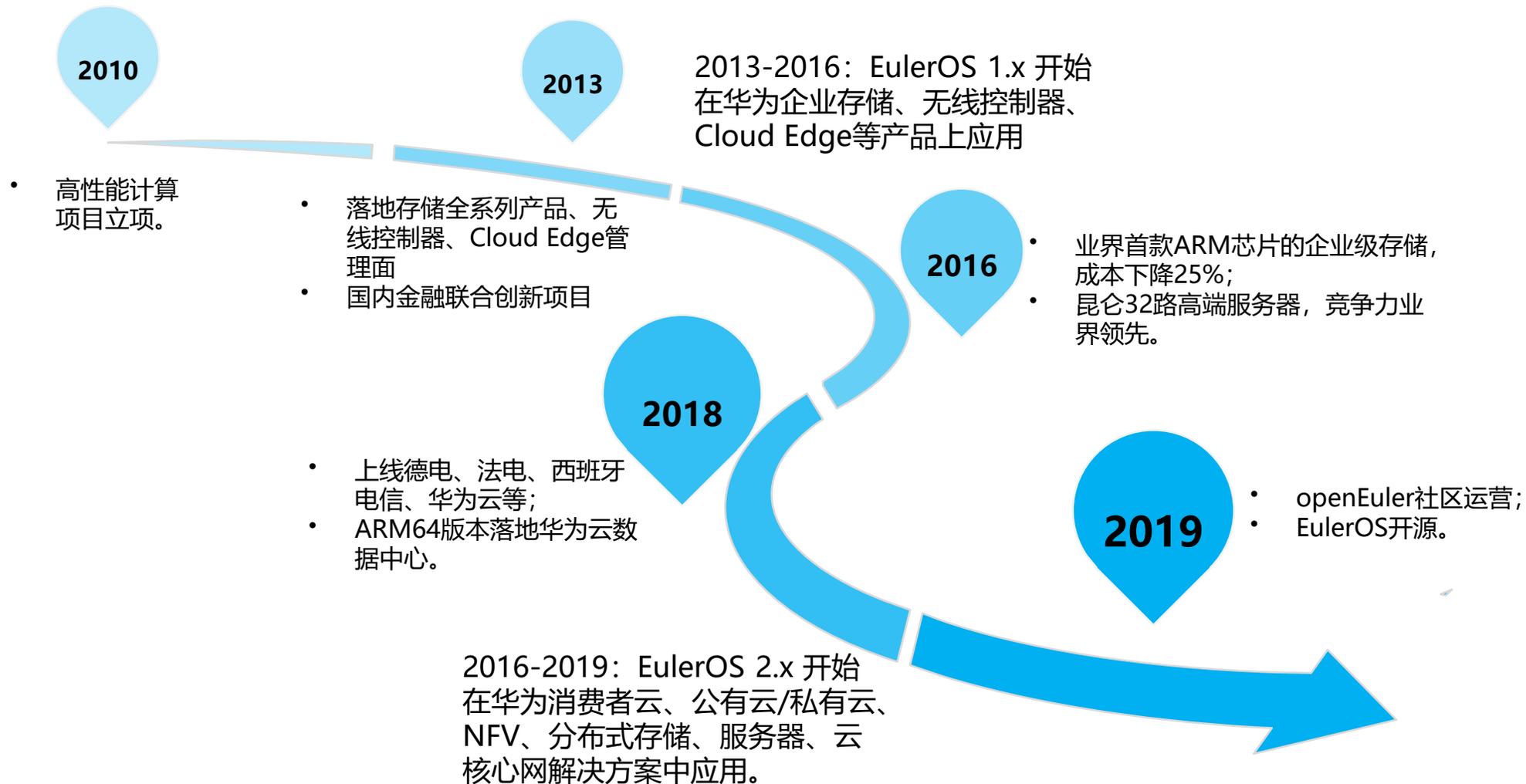


Dec 31, 2019

全面开放开源



openEuler发展历程





openEuler技术特性 - iSula容器解决方案 (1)

- iSula 通用容器引擎 (iSulad) 是一种新的容器解决方案, 具有以下特点:
 - 快速灵活&轻量级
 - 可信&安全启动
 - 升级不中断业务
 - 增强安全性和调测特性

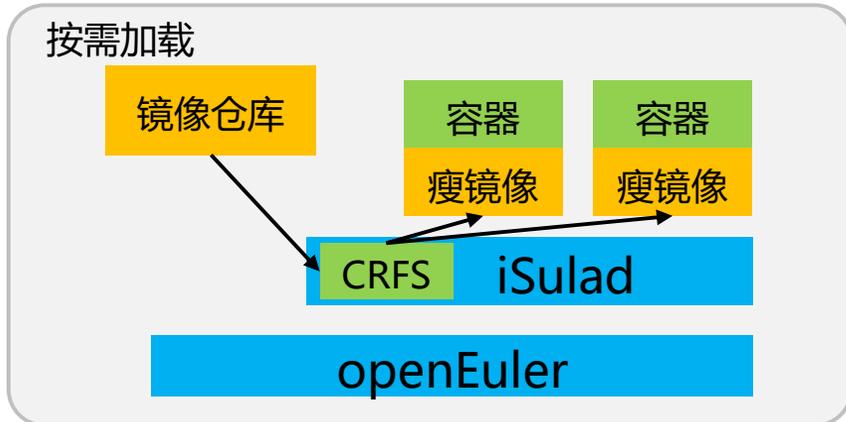
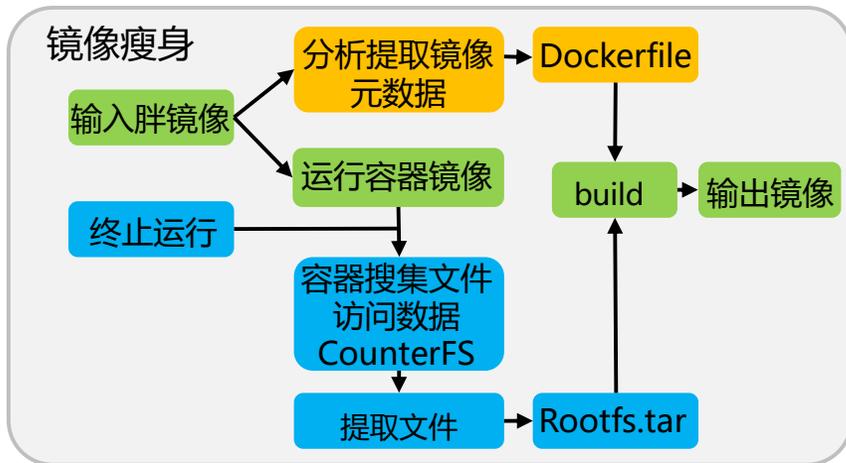


iSulad

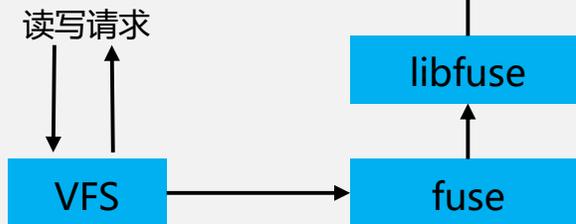


openEuler技术特性 - iSula容器解决方案 (2)

- iSula 容器基础镜像支持按需剪裁，实现极致小型化。



CounterFS实现架构



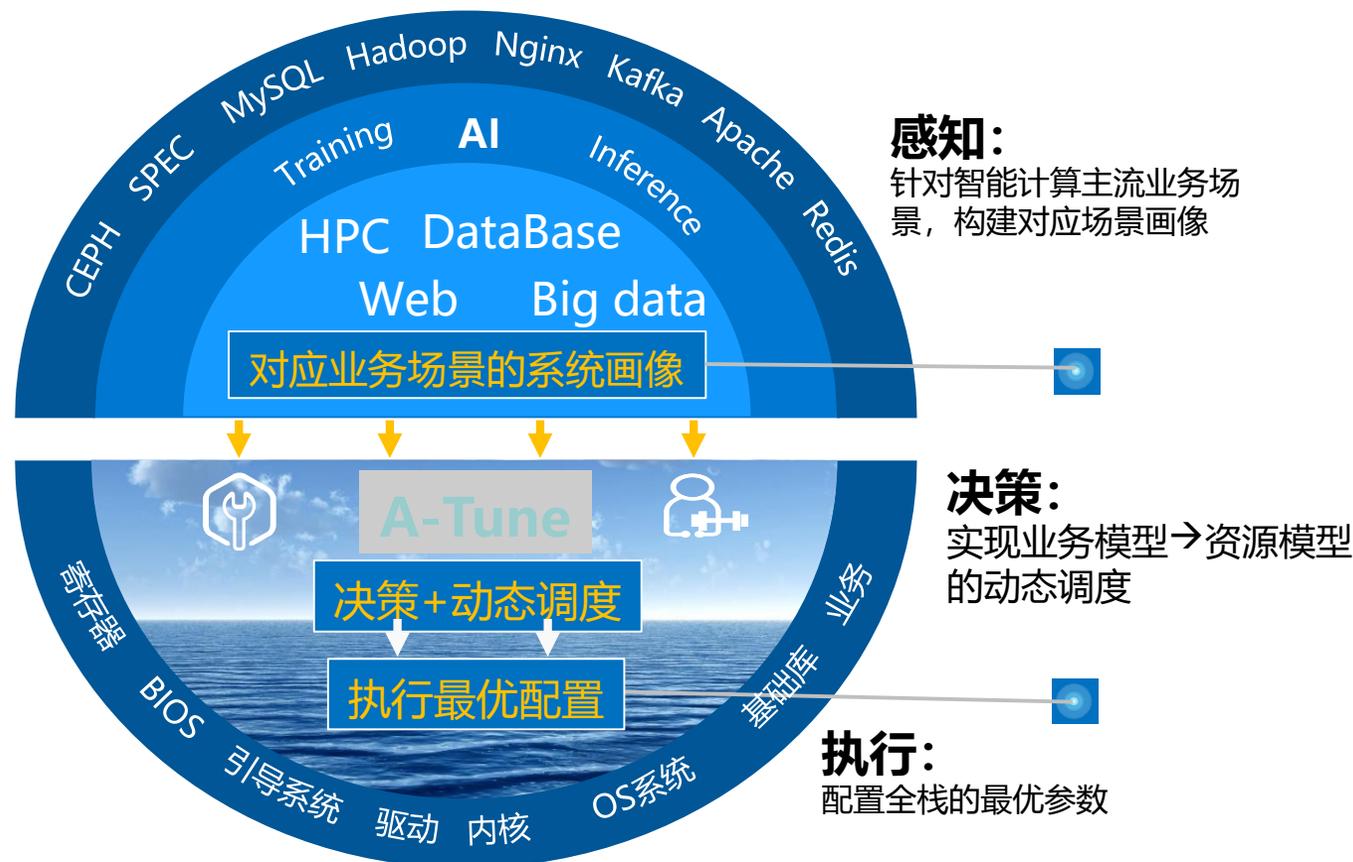
- 静态分析和动态分析相结合，只提取容器运行必须的文件
- 精简后的小型化容器镜像运行缺少的文件可通过端云协同的方案按需下载
- 容器镜像裁剪效果：

镜像名字	剪裁前大小	剪裁后大小
httpd(官方镜像)	178MB	7MB
nginx(官方镜像)	109MB	9MB



openEuler技术特性 - A-Tune资源调优自动化

- A-Tune是一种通过非侵入式系统画像的负载感知方法，识别业务并匹配最佳资源模型，实时响应业务特征变化的AI自动调优系统。





openEuler其他技术特性概览

系统安装

openEuler 1.0 改进了安装程序 Anaconda, 并做了个性化定制和修改, 方便用户安装。

定制/剪裁

openEuler 1.0 提供优化的 kiwi 工具对系统进行个性化的自定义修改。

文件系统

openEuler 1.0 中使用的默认文件系统是 ext4, 兼容 ext3/2 文件系统。openEuler 1.0 支持高性能日志文件系统 XFS。

系统管理

openEuler 1.0 使用 systemd 进行系统和服务的管理, systemd 与 SysV 和 Linux 标准的 init 脚本兼容。



系统安全

openEuler 1.0 提供多重安全手段, 包括身份识别与认证、安全协议、强制访问控制、完整性保护、安全审计等安全机制, 保障操作系统的安全性, 为各类上层应用提供安全基础。

系统调测

openEuler 1.0 支持 Perf 性能调测工具和 blktrace 块设备 I/O 跟踪工具。

编译器

openEuler 1.0 基于原生 GCC 进行了 Bug 修复、特性增强; 同时 openEuler 1.0 提供 Huawei Open JDK 支持, 相比较 Open JDK, Huawei Open JDK 作了优化、稳定性增强、安全增强等。

内核

openEuler 1.0 采用 kernel 版本 4.19。Linux Kernel 4.19 是一个 LTS 版本, 提供长期支持并持续更新版本, 以保障用户的 Linux 操作系统安全、可靠。



openEuler开源社区

欢迎加入openEuler社区

openEuler通过社区合作，打造创新平台，构建支持多处理器架构、统一和开放的操作系统，推动软硬件应用生态繁荣发展。

01

开发者

- openEuler社区采用gitee作为开发系统。
- 开发者可以在开源社区提交bug、参与讨论、贡献代码文案，也可以为openEuler的技术发展提出宝贵的建议，共同建设openEuler的繁荣生态

02

文档

- <https://openeuler.org>提供openEuler的所有文档，包括发行说明、安装指南、管理员指南，以及openEuler关键技术描述等文档。任何人均可以在<https://openeuler.org>上查看相关文档，学习openEuler。

03

下载

- <https://openeuler.org>提供openEuler相关操作系统的下载，有需求的玩家可以在开源社区下载网页下载openEuler相关操作系统。

04

社区会员

- openEuler开源社区欢迎各公司或者组织加入社区的一员，共同维护和管理社区，共同打造操作系统生态圈。



思考题

1. 鲲鹏计算平台已经兼容的操作系统可通过
<http://support.huawei.com/onlinetoolweb/ftca/index?serise=9> 链接查询。 ()



本节小结

- 本节在介绍了鲲鹏计算平台操作系统（OS）的兼容性，openEuler操作系统以及开源社区概况。



本章总结

- 本章详细介绍了鲲鹏计算产业以及鲲鹏生态，包括华为鲲鹏处理器，TaiShan服务器，华为云鲲鹏云服务以及鲲鹏计算平台兼容的操作系统等内容。
- 通过本章的学习，需掌握：
 - 鲲鹏计算产业的组成
 - 鲲鹏生态的概念
 - 华为鲲鹏处理器的型号、技术创新、应用场景
 - TaiShan服务器的规格型号
 - 华为云鲲鹏云服务的规格和应用场景
 - 鲲鹏计算平台的操作系统兼容性情况



学习推荐

- 《TaiShan 200服务器加速引擎开发者指南》

<https://support.huawei.com/enterprise/zh/doc/EDOC1100098461?idPath=7919749%7C9856522%7C9856629%7C23025673>

- openEuler开源社区：<https://openeuler.org>

- 鲲鹏社区：<https://www.huaweicloud.com/kunpeng/>



谢谢

www.huawei.com



应用移植



前言

- 本章将介绍了华为鲲鹏平台应用移植的相关知识。包括程序运行原理和软件迁移至鲲鹏计算平台的整个实施过程。并从服务器和容器两种应用载体出发，介绍了Kunpeng Porting Advisor迁移工具的使用和容器迁移操作步骤。



目录

1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
3. 容器迁移指导
4. 迁移常见问题及解决思路



本节概述和学习目标

- 本节将从软件迁移原理开始介绍，以了解软件迁移的背景与必要性。并通过软件迁移过程介绍了解迁移的方法以及流程。
- 学完本节后，您将能够：
 - 了解程序运行原理
 - 了解软件迁移至鲲鹏计算平台的过程



目录

1. 软件迁移原理和迁移过程

- 软件迁移原理概述
- 软件迁移过程概述
- 软件迁移典型案例

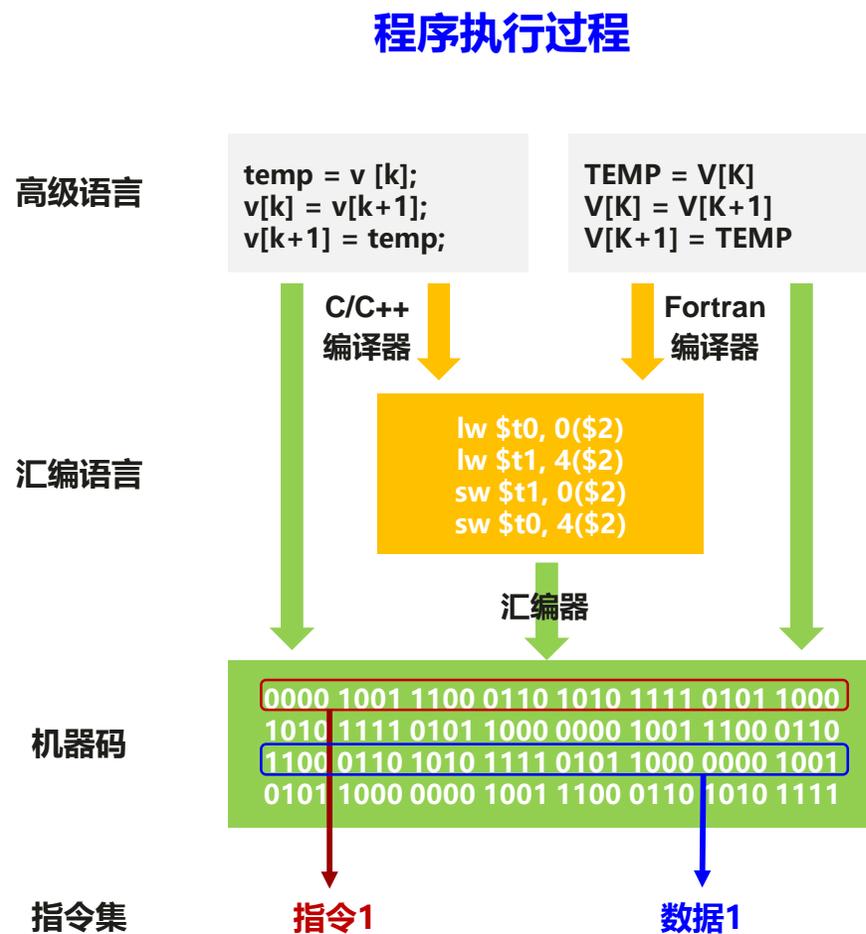
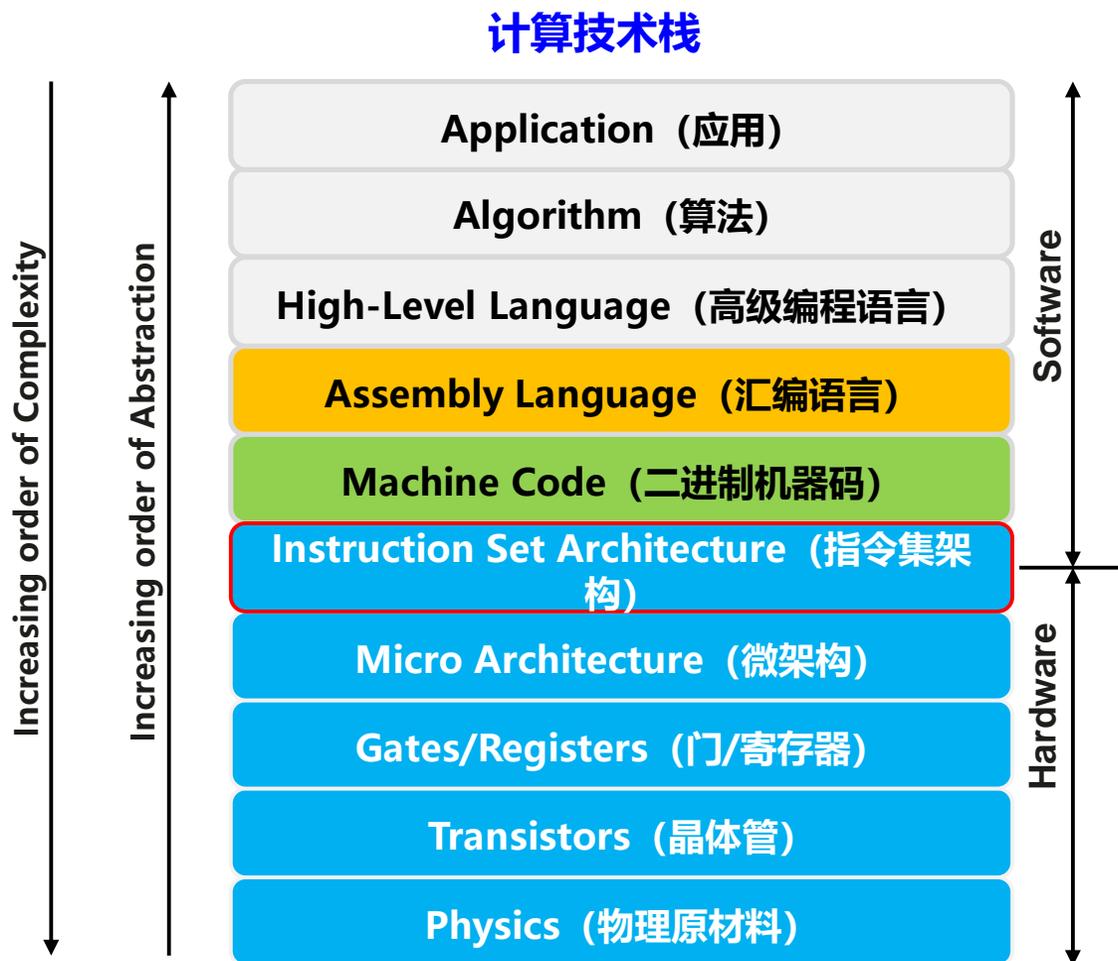
2. 迁移工具和迁移指导

3. 容器迁移指导

4. 迁移常见问题及解决思路



计算技术栈与程序执行过程





鲲鹏处理器与x86处理器的指令差异

程序代码 (C/C++) :

```
int main()
{
    int a = 1;
    int b = 2;
    int c = 0;

    c = a + b;

    return c;
}
```

编译

鲲鹏处理器指令

指令	汇编代码	说明
b9400fe1	ldr x1, [sp,#12]	从内存将变量a的值放入寄存器x1
b9400be0	ldr x0, [sp,#8]	从内存将变量b的值放入寄存器x0
0b000020	add x0, x1, x0	将x1(a)中的值加上x0(b)的值放入x0寄存器
b90007e0	str x0, [sp,#4]	将x0寄存器的值存入内存 (变量c)

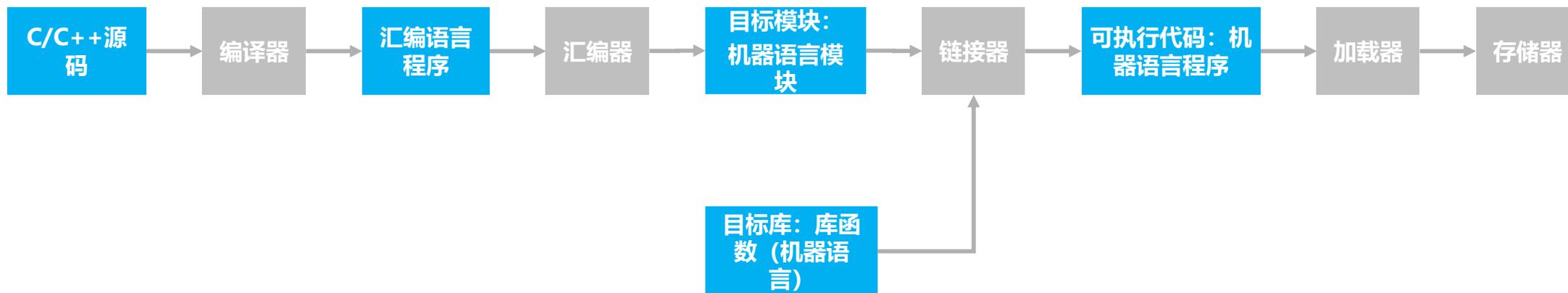
x86处理器指令

指令	汇编代码	说明
8b 55 fc	mov -0x4(%rbp),%edx	从内存将变量a的值放入寄存器edx
8b 45 f8	mov -0x8(%rbp),%eax	从内存将变量b的值放入寄存器eax
01 d0	add %edx,%eax	将edx(a)中的值加上eax(b)的值放入eax寄存器
89 45 f4	mov %eax,-0xc(%rbp)	将eax寄存器的值存入内存 (变量c)



从源码到可执行程序 - 编译型语言

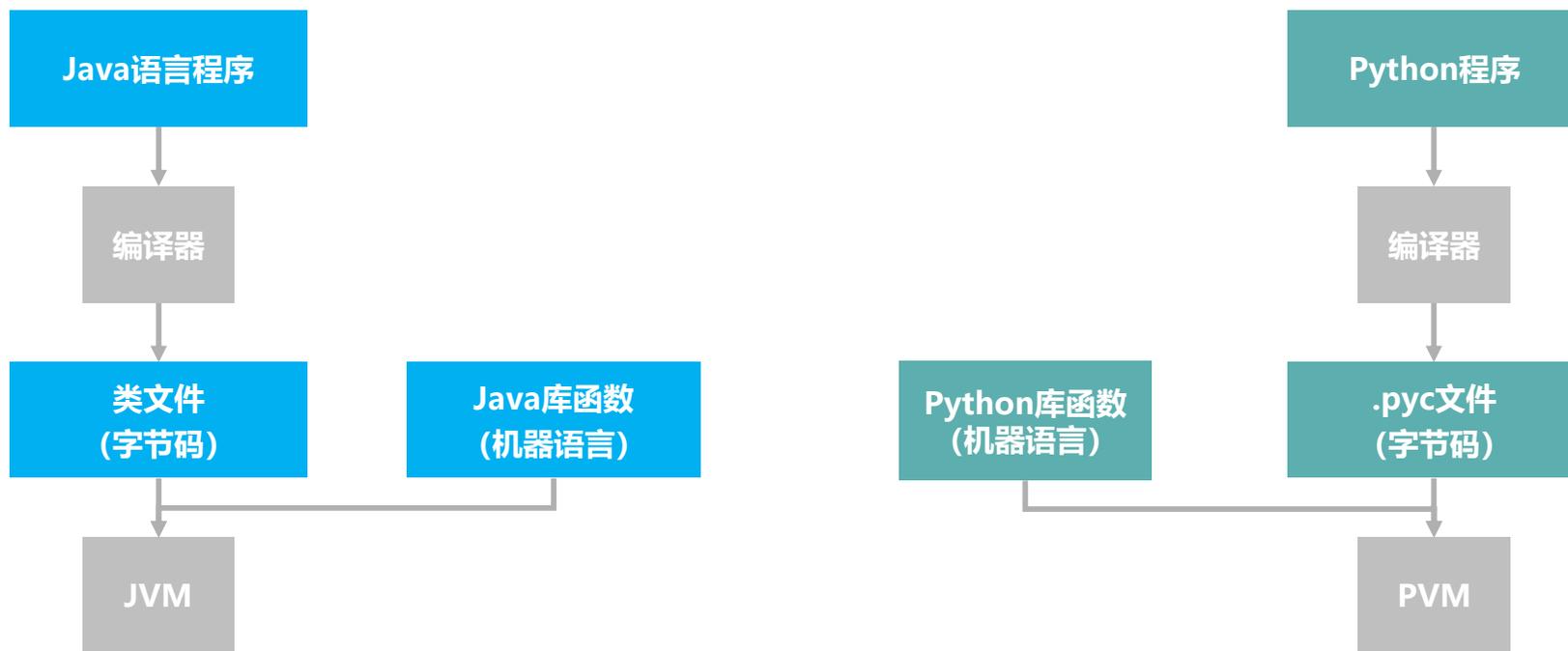
- **编译型语言**：典型的如C/C++/Go语言，都属于编译型语言。编译型语言开发的程序在从x86处理器迁移到鲲鹏处理器时，必须经过重新编译才能运行。
- 从源码到程序的过程：源码需要由编译器、汇编器翻译成机器指令，再通过链接器链接库函数生成机器语言程序。机器语言必须与CPU的指令集匹配，在运行时通过加载器加载到内存，由CPU执行指令。





从源码到可执行程序 - 解释型语言

- **解释型语言**：典型的如Java/Python语言，都属于解释型语言，解释型语言开发的程序在迁移到鲲鹏处理器时，一般不需要重新编译。
- 解释型语言的源代码由编译器生成字节码，然后再由虚拟机解释执行。虚拟机将不同CPU指令集的差异屏蔽，因此解释型语言的可移植性很好。但是如果程序中调用了编译型语言所开发的so库，那么这些so库需要重新移植编译。





目录

1. 软件迁移原理和迁移过程

- 软件迁移原理概述
- 软件迁移过程概述
- 软件迁移典型案例

2. 迁移工具和迁移指导

3. 容器迁移指导

4. 迁移常见问题及解决思路



迁移过程概述 - 五个阶段完成软件迁移

软件迁移 的五个阶段

阶段一
技术分析

阶段二
编译迁移

阶段三
功能验证

阶段四
性能调优

阶段五
规模商用

软件移植

- 软件栈分析 (应用软件、OS、数据库、中间件组件等)
- 编程语言/代码、依赖库分析

- 重写汇编代码
- 修改编译选项
- 代码编译 (含依赖库替换)

- 全量功能验证
- 交付工具适配

- 产品关键性能指标测试和调优
- 全面性能测试和调优

- 可靠性、可服务性验证和配置工具开发
- 上市资料刷新

迁移环境

- 准备调试编译环境 (准备测试样机服务器/OpenLab线上服务器)

- 搭建编译调试环境 (OS/编译器/CI工程等)

- 搭建功能测试环境

- 部署测试工具

- 部署生产系统
- 灰度/割接上线

管理工作

- 成立项目组
- 制定迁移计划
- 协调相关人力/物料资源

- 例行监控与沟通汇报

- 例行监控与沟通汇报

- 例行监控与沟通汇报

- 项目总结关闭



阶段一：软件移植 - 软硬件技术栈整体迁移策略



软件技术栈

业务软件

开源软件

- **开源软件**：获取开源软件ARM64软件包或下载源码重新编译
- **依赖组件**：获取ARM64包或获取源码重新编译，闭源组件需升级替换ARM版本或替换类似组件库

自研软件

- **编译语言**：C/C++/Go类编译语言重新编译
- **解释语言**：Java/Python等解释型语言替换ARM版本JDK或PVM虚拟机

商用软件

- 获取支持鲲鹏芯片或TaiShan服务器的软件版本
- 如无法获取到兼容版本，需更换其他类似软件

JDK/PVM

- 替换支持鲲鹏、TaiShan版本，如OpenJDK、Huawei JDK等

编译器

- 替换支持鲲鹏、TaiShan版本编译器，如GCC 9.1以上

操作系统

- 替换支持鲲鹏、TaiShan版本操作系统，可参考软件兼容性清单

驱动

- 替换支持鲲鹏、TaiShan版本的驱动

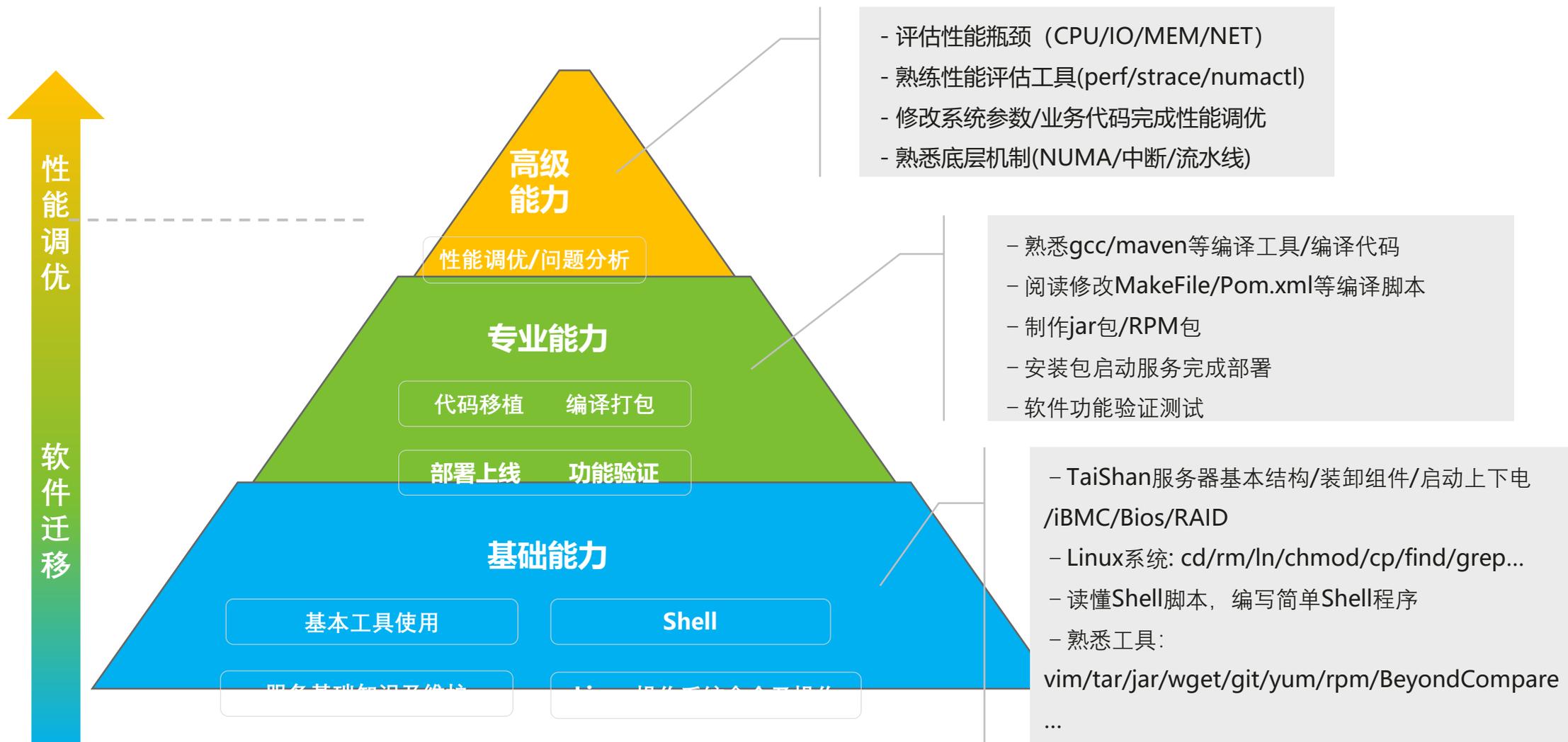
迁移环境

- 获取Taishan服务器测试样机/或申请OpenLab线上服务器资源

说明：在华为云社区的鲲鹏论坛，已经建立了软件仓库，部分软件包可以在软件仓库直接下载



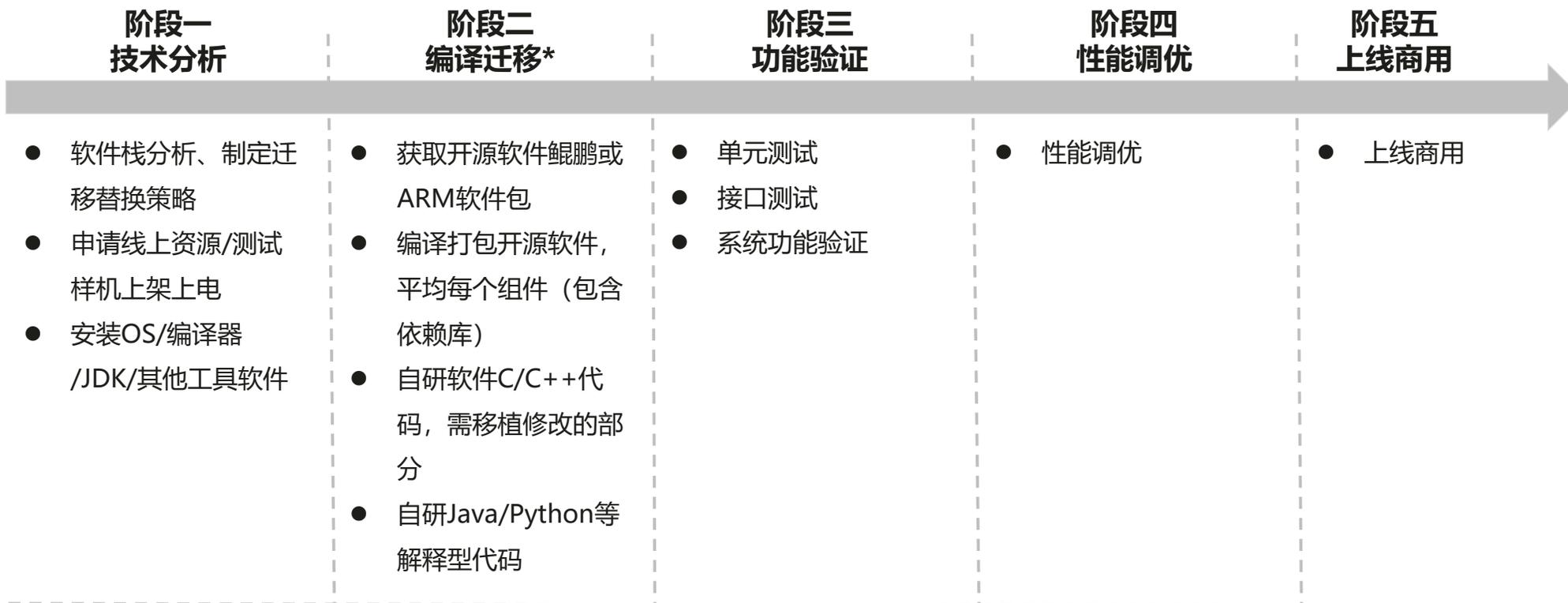
阶段一：管理工作 - 软件迁移团队能力模型与要求





阶段一：管理工作 - 制定迁移计划

主要活动





阶段一：迁移环境 - 准备迁移环境

通过华为云鲲鹏云服务器搭建开发环境

1. 登录华为云官网，查看弹性云服务器<https://www.huaweicloud.com/product/ecs.html>
2. 通过华为云账号登录，点击立即购买
3. CPU架构选择鲲鹏计算，然后按照指导完成鲲鹏弹性云服务器的购买



计费模式: 包年/包月 按需计费

区域: 华北-北京四

可用区: 随机分配 可用区1 可用区2 可用区3

CPU架构: X86计算 鲲鹏计算

规格: 最新系列 vCPUs 全部 内存 全部 规格名称

鲲鹏通用计算增强型 鲲鹏内存优化型

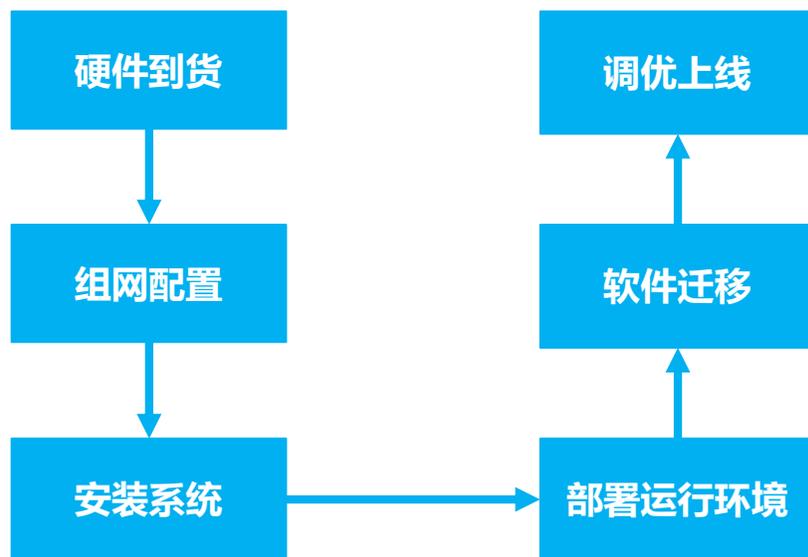
规格名称	vCPUs 内存	CPU
<input checked="" type="radio"/> kc1.small.1	1vCPUs 1GB	Huawei Kunpeng 920 2.6GHz
<input type="radio"/> kc1.large.2	2vCPUs 4GB	Huawei Kunpeng 920 2.6GHz
<input type="radio"/> kc1.large.4	2vCPUs 8GB	Huawei Kunpeng 920 2.6GHz
<input type="radio"/> kc1.xlarge.2	4vCPUs 8GB	Huawei Kunpeng 920 2.6GHz
<input type="radio"/> kc1.xlarge.4	4vCPUs 16GB	Huawei Kunpeng 920 2.6GHz
<input type="radio"/> kc1.2xlarge.2	8vCPUs 16GB	Huawei Kunpeng 920 2.6GHz
<input type="radio"/> kc1.2xlarge.4	8vCPUs 32GB	Huawei Kunpeng 920 2.6GHz
<input type="radio"/> kc1.3xlarge.2	12vCPUs 24GB	Huawei Kunpeng 920 2.6GHz
<input type="radio"/> kc1.3xlarge.4	12vCPUs 48GB	Huawei Kunpeng 920 2.6GHz



阶段一：迁移环境 - 准备迁移环境

本地TaiShan服务器环境

● 本地迁移流程



- 服务器要连接公网环境
- 配置物理服务器，组网，需要一定周期
- 仅本地可用，无认证



阶段一：迁移环境 - 准备迁移环境

远程Openlab

通过开放工具化、流程化能力，支持面向全软件栈的鲲鹏生态建设

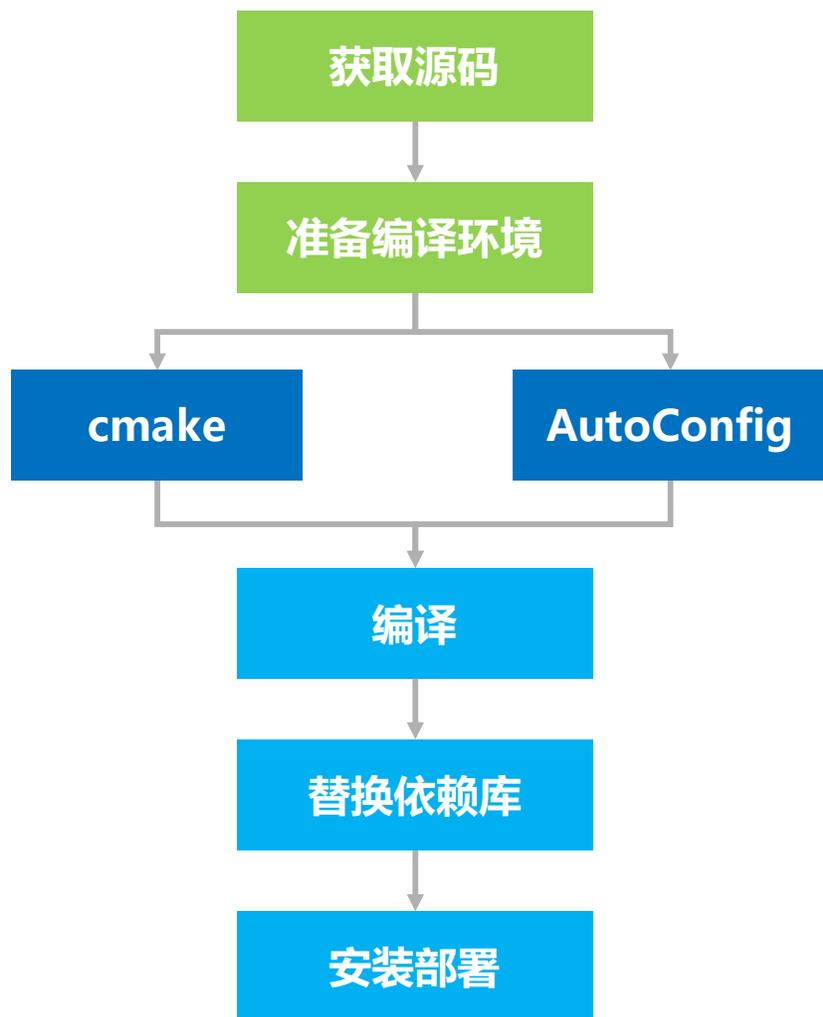


业务流	源码移植检查	编译构建	性能调优	自动化检测	认证发布
测试套	兼容性测试	稳定性测试	性能测试	安全测试	功耗测试
工具链	移植知识库	扫描工具	移植工具	性能分析工具	自动化检测工具

- 办公环境可以连接公网环境
- 无需配置物理服务器，向Openlab申请远程服务器资源
- 提供认证，生态推广



阶段二：开源软件源码编译移植（C/C++代码）

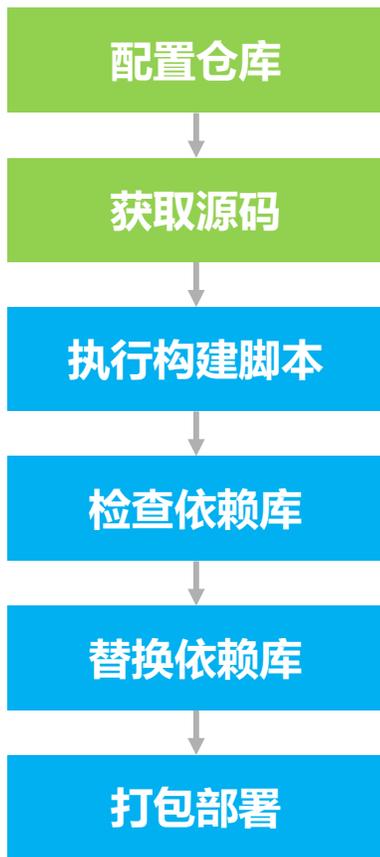


- 1、获取源码：通过github或第三方开源社区获取源码；
- 2、准备编译环境：安装编译器gcc等；
- 3、使用开源软件源码中的cmake或AutoConfig脚本生成makefile；
- 4、执行makefile编译可执行程序；
- 5、替换依赖库：通过开源软件readme文件、编译时的so库缺失或链接错误，重新编译或替换依赖库；
- 6、将可执行程序安装部署到生产或测试系统。

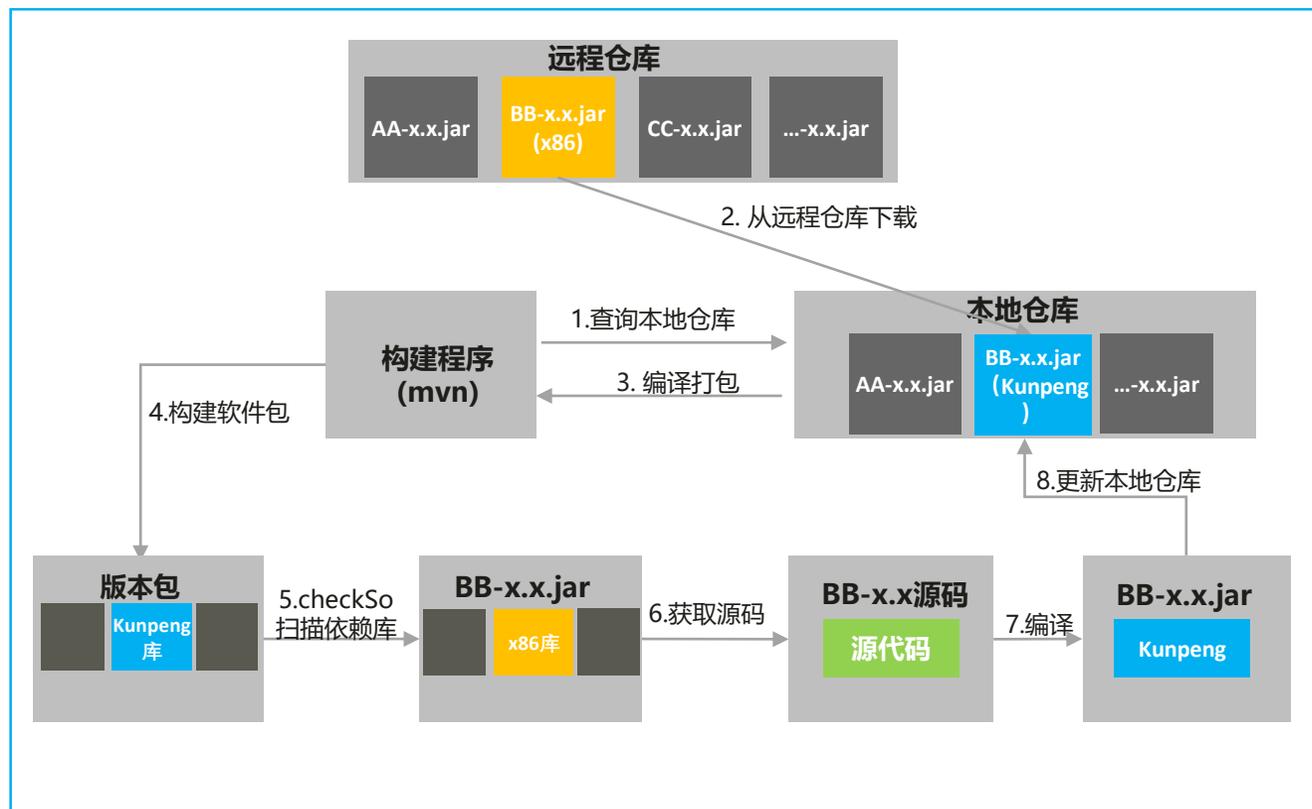


阶段二：开源软件Maven仓库软件迁移 (Java)

开源软件迁移过程 (Maven软件仓库)



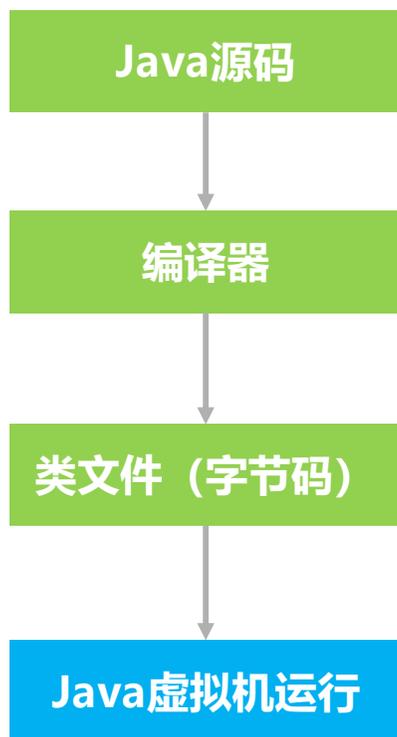
Maven仓库软件构建流程 (Maven软件仓库)





阶段二：Java自研代码移植

Java程序编译运行的过程



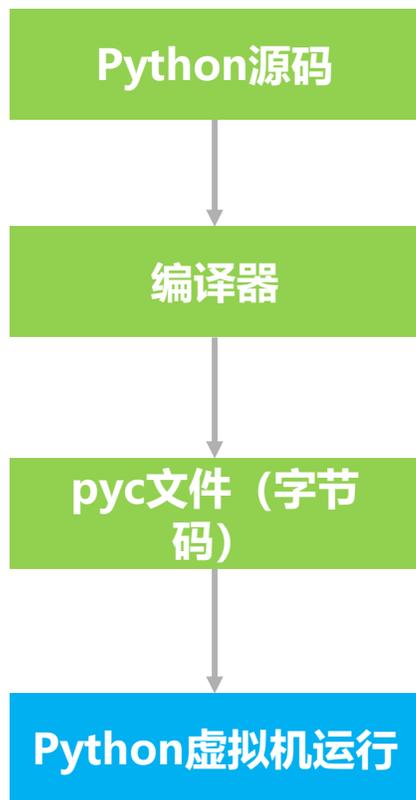
Java程序迁移至鲲鹏处理器

- 1、安装鲲鹏或ARM版本JDK
- 2、配置JDK环境变量
- 3、编译Java源码生成字节码
- 4、【可选】移植jar包中的依赖库
(替换鲲鹏/ARM版本或获取C/C++源码重新编译)
- 5、【可选】使用新的依赖库重新打jar包
- 6、启动Java程序，调试功能



阶段二：Python自研代码移植

Python程序编译运行的过程



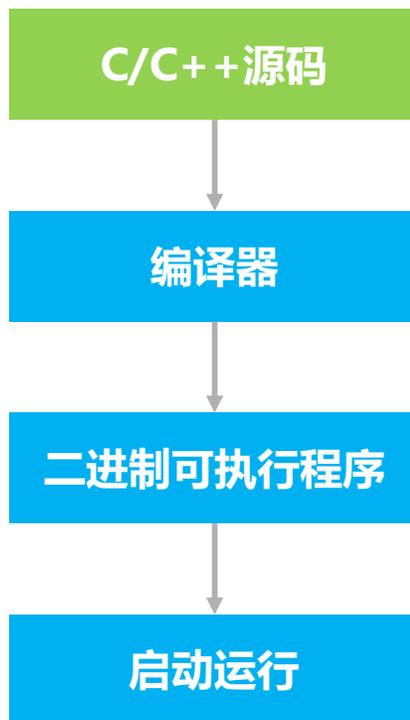
Python程序迁移至鲲鹏处理器运行

- 1、使用操作系统自带的Python，或安装兼容鲲鹏/ARM版本的Python软件
- 2、设置Python执行环境变量
- 3、【可选】编译生成pyc文件（字节码）
(替换鲲鹏/ARM版本或获取C/C++源码重新编译)
- 4、【可选】移植外部依赖库
- 5、【可选】更新代码中新的外部依赖库调用代码
- 6、执行Python源码程序



阶段二：C/C++ 自研代码移植

C/C++ 程序编译运行的过程



C/C++ 程序迁移至鲲鹏处理器

- 1、安装编译器（推荐GCC 7.3.0以上版本）
- 2、配置GCC环境变量
- 3、修改C/C++源码
- 4、编译C/C++源码，生成可执行程序
- 5、启动C/C++程序，调试功能



阶段二：C/C++代码builtin函数、数据类型移植

功能说明

X86代码

鲲鹏代码

数据类型

显示定义char类型
变量为有符号型

```
char a = 'a'
```

```
signed char a = 'a'
```

Builtin函数

使用编译器自带的
builtin函数

```
__builtin_ia32_crc32qi (__a, __b);
```

```
__builtin_aarch64_crc32b (__a, __b);
```

把内存数据预取到
cache中

```
asm volatile("prefetch0 %0" :: "m"  
(*(unsigned long *)x));
```

```
#define prefetch(_x)  
__builtin_prefetch(_x)
```



阶段二：C/C++ 代码编译选项、编译宏移植

	功能	X86编译选项	鲲鹏编译选项
64位编译	定义编译生成的应用程序为64位，编译选项不同	-m64	-mabi=lp64
ARM指令集	Makefile中定义指令集类型，由X86修改为ARM	-march=broadwell	-march=armv8-a
编译宏	原有X86版编译宏替换成ARM版	__X86_64__	__ARM_64__



阶段二：其他小语种代码移植

Go

lua

Ruby

PHP

Scala

Perl

TypeScript

JavaScript

编译型

解释型

编译型：源码需要重新编译

解释型：源码不用编译，安装解释器即可

搭建程序运行环境方法：

- 1、操作系统自带解释器软件，直接安装即可使用；
- 2、从官网上下载支持ARM的解释器软件包，直接解压使用；
- 3、从官网下载解释器源码，编译后使用。



阶段二：商用闭源软件迁移（商用数据库软件）





目录

1. 软件迁移原理和迁移过程

- 软件迁移原理概述
- 软件迁移过程概述
- 软件迁移典型案例

2. 迁移工具和迁移指导

3. 容器迁移指导

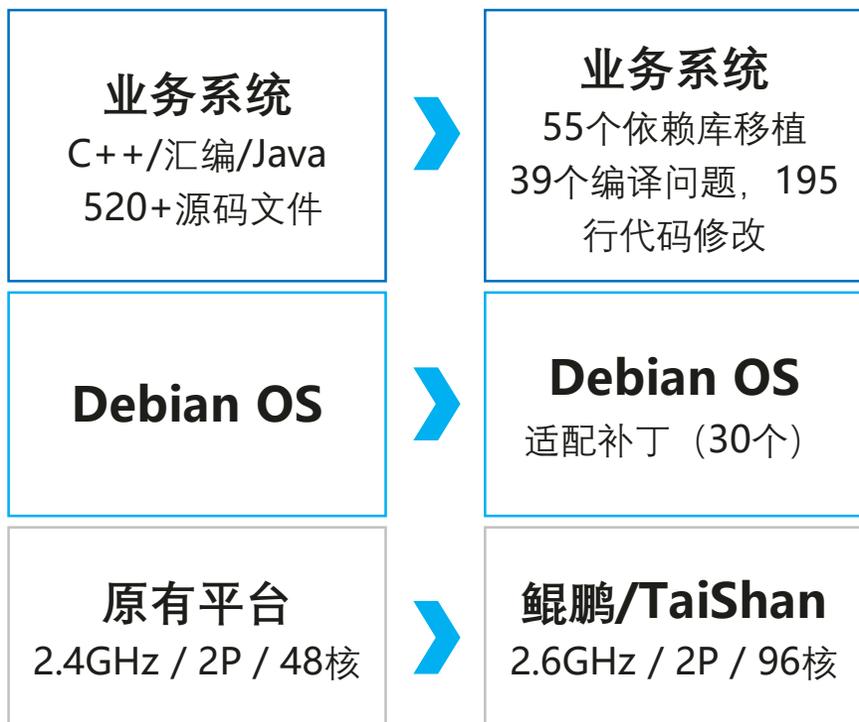
4. 迁移常见问题及解决思路



某行业伙伴快速实现软件迁移的案例

案例：行业伙伴核心应用系统迁移

5人月完成520+源码文件移植，性能提升56%



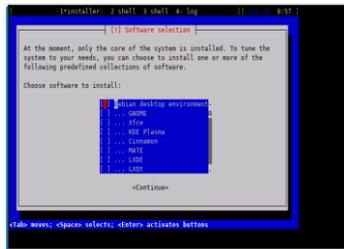
环境准备：1人月

代码编译和修改：2人月

性能优化：2人月

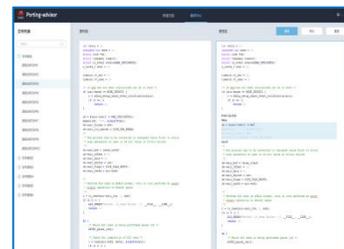
技术分析

硬件环境
代码评估
环境部署



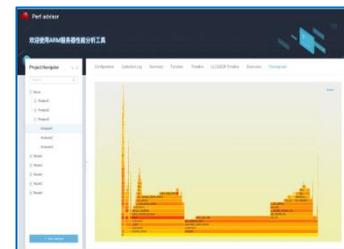
代码移植

C++/汇编语言移植
安装JDK
编译开源代码
更换依赖库



性能优化

热点函数分析
多核绑定
内存就近访问
网卡队列优化





思考题

1. 为什么x86架构处理器上的软件在鲲鹏处理器使用时需要移植？（ ）
 - A. 两种处理器的指令集不同
 - B. 源代码需要按照目标处理的指令集编译成指令才能运行
 - C. 编译型语言由编译器静态编译成指令和数据
 - D. 解释型语言由语言的虚拟机在运行时将源码/字节码编译成指令和数据



本节小结

- 本节概述了迁移的原理，为什么需要进行软件迁移，软件迁移所具备的环境是什么，以及软件迁移的实施方法与流程。



目录

1. 软件迁移原理和迁移过程
- 2. 迁移工具和迁移指导**
3. 容器迁移指导
4. 迁移常见问题及解决思路



本节概述和学习目标

- 上一节中我们讲述了软件迁移的原理和实施过程，本节将通过学习华为鲲鹏代码迁移工具，掌握迁移工具的应用场景和操作过程。
- 学完本节后，您将能够：
 - 了解华为鲲鹏代码迁移工具的定位和应用场景
 - 掌握C/C++类应用程序移植的操作过程
 - 掌握Java类应用程序移植的操作过程



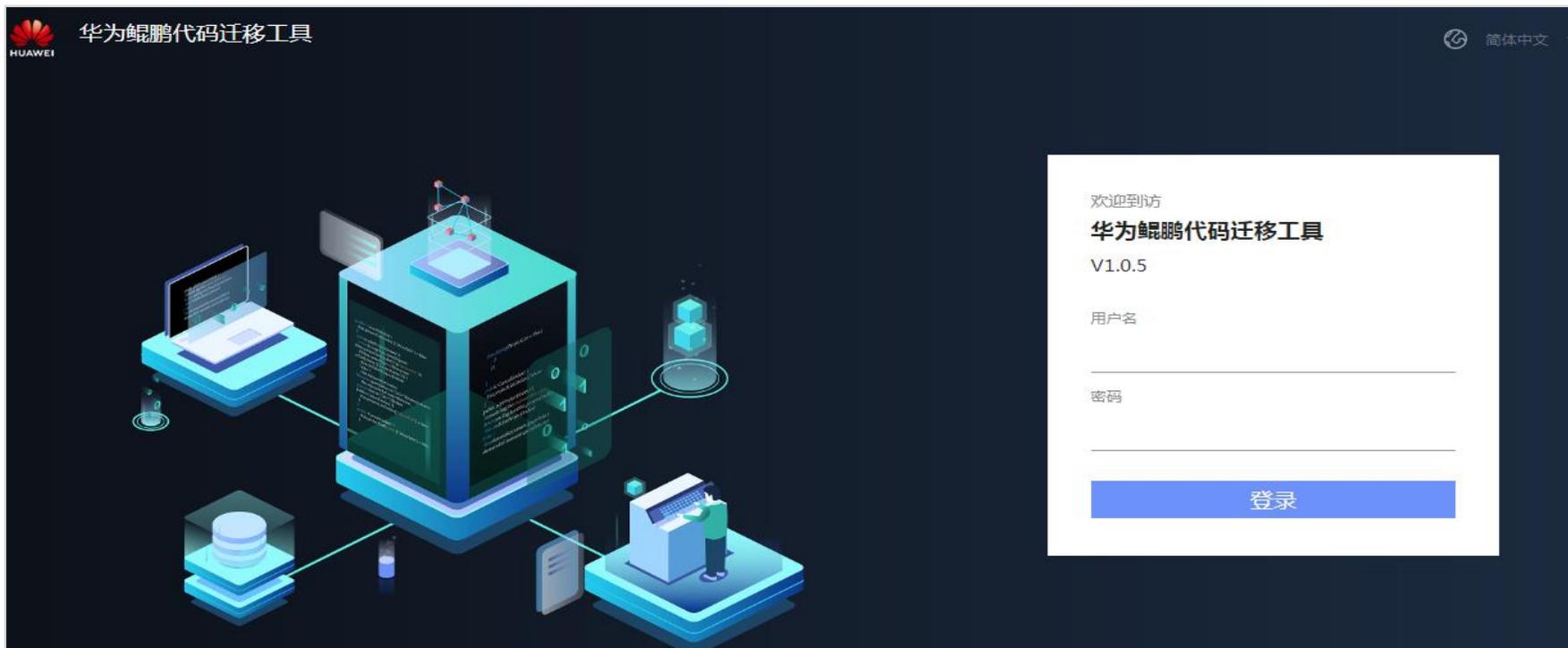
目录

1. 软件迁移原理和迁移过程
- 2. 迁移工具和迁移指导**
 - 华为鲲鹏代码迁移工具介绍
 - C/C++类应用移植
 - Java类应用移植
3. 容器迁移指导
4. 迁移常见问题及解决思路



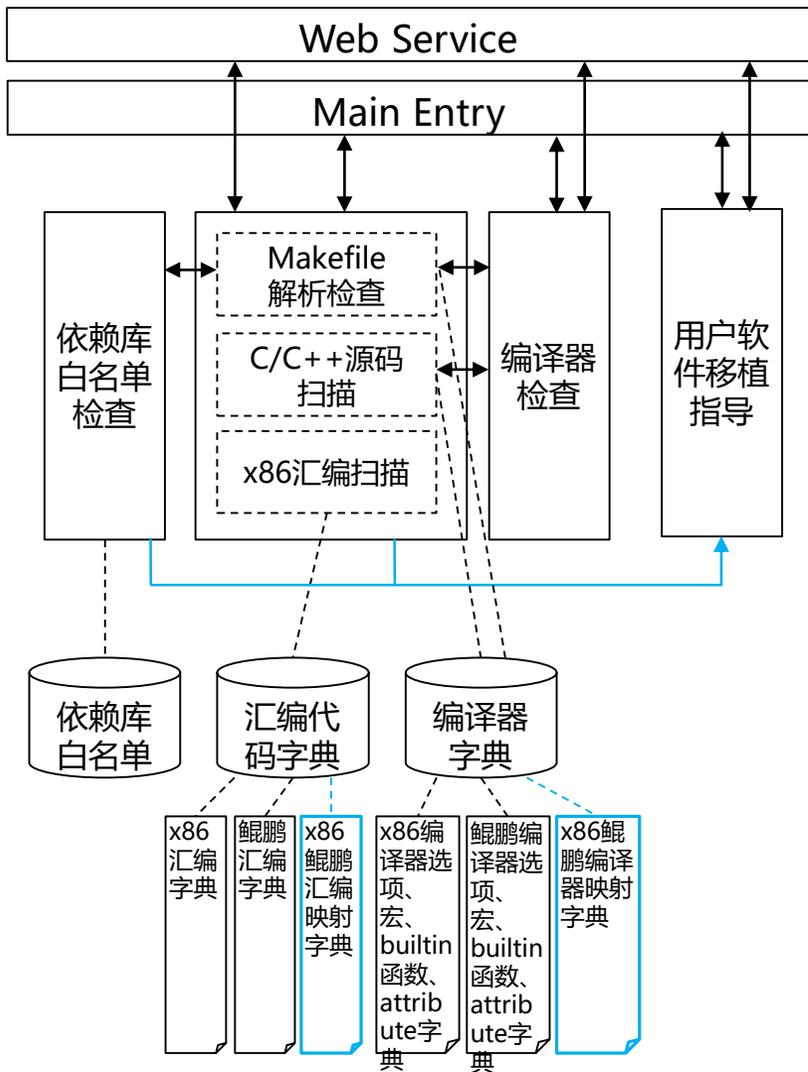
华为鲲鹏代码迁移工具是什么

- **华为鲲鹏代码迁移工具**主要面向鲲鹏平台的开发者、用户和第三方待移植软件提供方开发工程师，用来分析待移植软件源码文件，并给出代码移植指导报告，同时能够自动分析出需修改的代码内容，并指导如何修改，帮助用户顺利完成应用从x86平台向鲲鹏平台的移植。





华为鲲鹏代码迁移工具逻辑架构



模块名称	功能说明
Web Service	Web服务器。
Main Entry	代码迁移工具命令行入口。
依赖库白名单检查	检查软件构建工程文件中的SO文件名列表，对比SO依赖库白名单，得到需要移植的SO依赖库的详细信息。
Makefile解析检查	检查软件构建工程文件中使用的链接库、编译选项、宏定义，并提供移植修改建议。
C/C++源码扫描	检查源码中使用编译器宏、builtin函数、attributes，并提供移植修改建议。
x86汇编扫描	检查源码中使用x86汇编，提供移植修改建议。
编译器检查	根据编译器版本确定x86平台与鲲鹏平台相异的编译器宏、编译选项、builtin函数、attribute函数等列表。
用户软件移植指导	<ol style="list-style-type: none"> 根据包扫描和C/C++源码扫描结果合成用户软件移植分析报告。 根据源码检查结果，结合软件替换规则字典，为每个修改点增加修改建议。



华为鲲鹏代码迁移工具业务流程



输入

源码文件:

- C/C++源代码文件
- 汇编源代码文件
- Makefile文件



分析处理

源码文件扫描分析:

- 用户C/C++需要移植部分识别
- 汇编源码同功能指令集、兼容指令集移植部分
- 从Makefile中识别需要移植/替代的编译依赖库
- 根据知识库给出移植指导建议



输出

报告文件:

- 分析文件、分析时间戳等信息
- 关键修改内容综述, 简单说明
- 详细csv或html报告, 需修改的代码行号, 更改点及指导建议
- 编译依赖库移植或替换建议



功能特性 (1)

华为鲲鹏代码迁移工具可以作为独立软件提供给用户安装使用，支持的功能特性如下：

- 检查用户软件C/C++软件构建工程文件，并指导用户如何移植该文件。

举例： Makefile文件检查结果中，原始文件中存在编译选项-m32，工具建议移植到鲲鹏平台时请删除该编译选项。

-m32是x86 32位应用编译选项， m32选项设置int为32bits及long、指针为32 bits，为AMD的x86架构生成代码。在鲲鹏平台无法支持。

```
46 #  
47 # CFLAGS  
48 #  
49 CFLAGS = -m32 -nostdlib -fno-builtin -W -Wall -Werror -O0  
50 #  
51 CFLAGS = -m32 -nostdlib -fno-builtin -W -Wall -Werror -O0
```

Suggestion: Please remove the '-m32' option on Kunpeng platform.
Description: Generate code for a 32-bit or 64-bit environment. The '-m32'

- 检查用户软件C/C++软件构建工程文件使用的链接库，并提供可移植性信息。

举例： 移植报告检查结果中，原始文件中存在软件构建工程文件使用的SO链接库，工具建议下载已有二进制安装包，然后上传至目标服务器进行安装使用即可。

序号	名称	描述	操作建议	操作
1	libz	Buffer compression library.	Download and install the installation package.	下载
2	libcrypto	Encyption/Decryption library.	Download and install the installation package.	下载
3	libssl	Secure socket library.	Download and install the installation package.	下载



功能特性 (2)

- 检查用户软件C/C++源码，并指导用户如何移植源文件。

举例：对于用户软件C/C++源码，工具主要检查编译器宏、builtin函数、attributes等，并提供移植建议。

- 1) 对于编译器宏中的_x86_64_字段，工具建议移植到鲲鹏平台时请删除该字段。
- 2) 对于使用编译器自带的builtin函数_builtin_ia32_crc32qi (_a, _b);，工具建议修改为_builtin_aarch64_crc32b (_a, _b);

- 检查用户软件中x86汇编代码，并指导用户如何移植。

举例：对于用户软件x86内存操作的汇编指令，需要从mov指令替换为ldr x0, [x1] & str x0, [x2]指令。CRC32C校验值计算的汇编指令，需要从crc32q指令替换为crc32cb、crc32ch、crc32cw、crc32cx指令。无条件跳转的汇编指令，需要从JMP指令替换为b指令。

- 支持命令行和Web两种访问方式。
- 支持SO依赖库白名单升级。



应用场景

含有源代码的软件从x86平台移植到鲲鹏平台

- 从x86平台移植到TaiShan服务器
- 从x86平台移植到华为鲲鹏云主机



部署方式

- 华为鲲鹏代码迁移工具采用单机部署方式，即将工具部署在用户的开发、生产环境的x86服务器或云服务器上。部署环境要求如下表所示。

类别	子类	要求
硬件	服务器	<ul style="list-style-type: none">• x86物理服务器• TaiShan 200服务器
虚拟机	服务器	<ul style="list-style-type: none">• 弹性云服务器ECS
操作系统	CentOS	<ul style="list-style-type: none">• CentOS 7.6 (x86、TaiShan服务器、云服务器)• openEuler 1.0 (TaiShan服务器)



如何访问和使用

华为鲲鹏代码迁移工具提供CLI和Web两种访问方式，安装时由用户选择，只能安装一种。

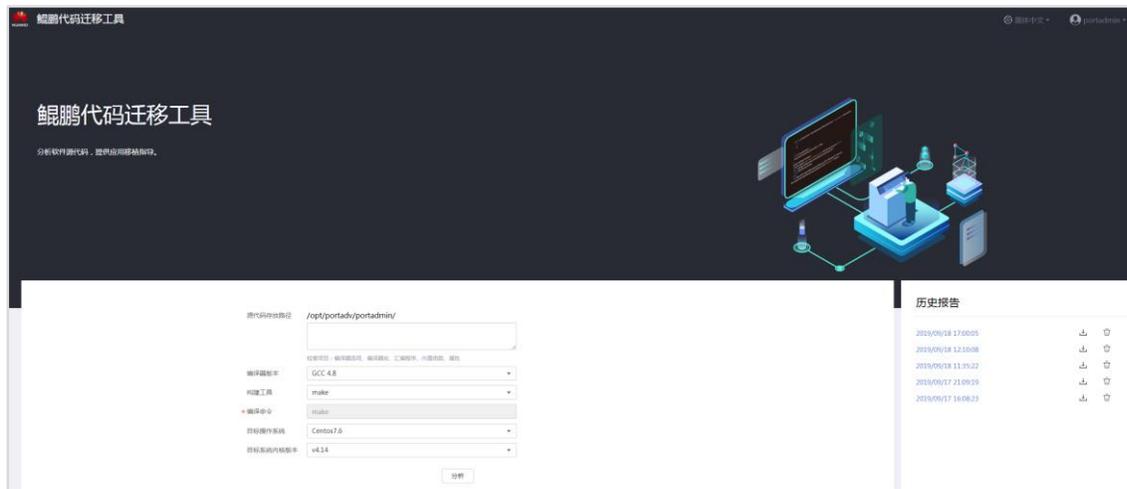
- CLI方式:

通过命令行方式使用代码迁移工具各功能，最终移植分析结果输出到.csv文件，用户可以根据移植建议进行处理。

```
[root@ecs-centos-test ~]# cd /opt/portadv/tools/bin/porting_advisor
```

- Web方式:

通过浏览器远程使用代码迁移工具各功能，最终移植分析结果输出到.csv或者.html文件中，用户可以根据移植建议进行处理。工具只允许一个用户工作，不支持多用户在线和并发访问。工具具有用户管理功能，由管理员创建和管理。每个用户需要创建自己的工作空间，每个用户的代码扫描任务都在各自的工作空间内完成。





目录

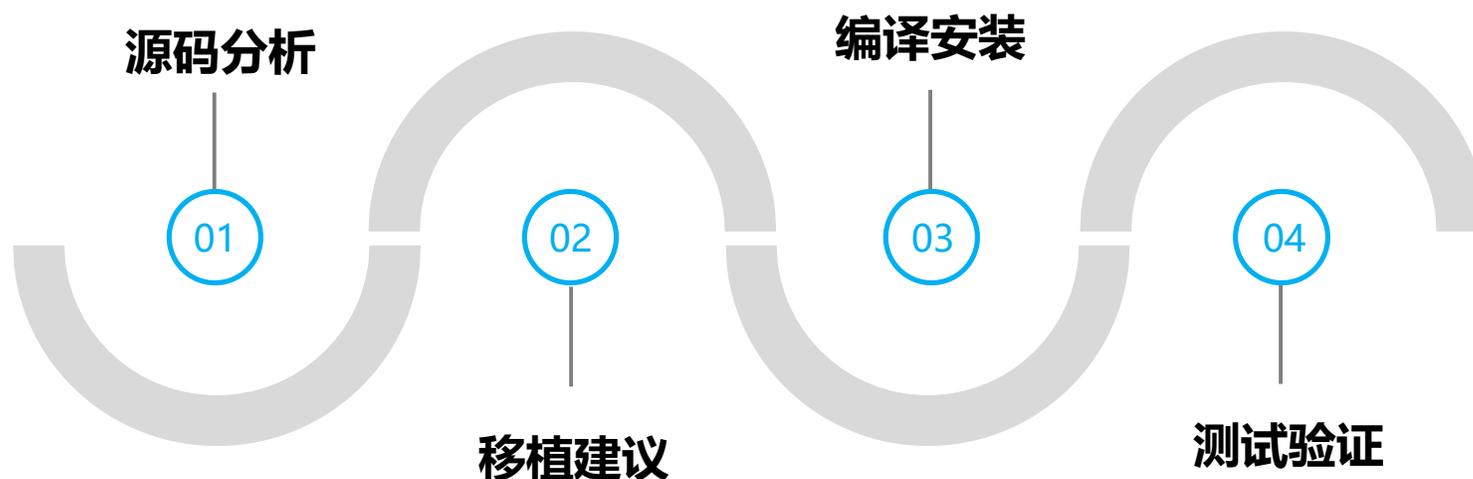
1. 软件迁移原理和迁移过程
- 2. 迁移工具和迁移指导**
 - 华为鲲鹏代码迁移工具介绍
 - C/C++类应用移植
 - Java类应用移植
3. 容器迁移指导
4. 迁移常见问题及解决思路



C/C++ 类应用移植

基于编译型语言开发的应用程序，例如C/C++语言应用程序，其编译后得到可执行程序，可执行程序执行时依赖的指令是CPU架构相关的。因此，基于x86架构编译的C/C++语言应用程序，无法直接在TaiShan服务器或华为鲲鹏云服务器上运行，需要进行移植编译。

应用移植的流程如下：





源码分析 (1)

- Step 1 登录代码迁移工具Web界面。
 - 在浏览器地址栏输入<https://部署服务器的IP:端口号>
例如: <https://10.116.239.242:8084>, 默认用户名为 **portadmin**, 默认密码为 **Admin@9000**
- Step 2 (可选) 如果需要使用其他用户登录Web界面进行移植分析时, 请单击界面右上角用户名下拉列表, 选择“用户管理”, 然后创建用户。创建用户后, 工具自动创建该用户的工作空间。
- Step 3 使用WinSCP工具将源码文件上传至代码迁移工具所在服务器的“/opt/portadv/*用户名*”路径下。
 - 例如管理员用户上传至“/opt/portadv/portadmin”下。

The screenshot illustrates the user management workflow in the code migration tool's web interface. It is divided into three main sections:

- Top Left (Dark Panel):** A sidebar menu for the user 'portadmin'. The '用户管理' (User Management) option is highlighted with a red box. Other options include '操作日志' (Operation Log), '修改密码' (Change Password), and '退出' (Logout).
- Top Right (User Management Page):** The '用户管理' (User Management) page is shown. It features a blue '创建用户' (Create User) button. Below it is a table with columns for '用户名' (Username), '角色' (Role), and '工作空间' (Workspace). The table is currently empty, showing '总条数: 0' (Total items: 0) and a page number '1'.
- Bottom (Create User Form):** An arrow points down to the '创建用户' (Create User) form. It contains the following fields:
 - * 用户名 (Username): A text input field with a note below it: '用户创建后名称不允许修改。' (Name cannot be modified after creation).
 - 工作空间 (Workspace): A dropdown menu currently showing '/opt/portadv/'.
 - * 管理员密码 (Admin Password): A text input field.
 - * 密码 (Password): A text input field.
 - * 确认密码 (Confirm Password): A text input field.



源码分析 (2)

- Step 4 使用默认管理员或新创建的普通用户登录代码迁移工具Web界面。
- Step 5 新建源码移植分析任务。参数说明如下表所示。
- Step 6 单击“分析”，立即启动源码可移植性分析。

参数名称	参数说明
源代码存放路径	源代码存放路径子目录。即Step3中上传的源码文件的路径。
编译器版本	选择编译器版本。当前版本只支持GCC 4.8。
构建工具	选择构建工具。当前版本只支持make。
编译命令	选择源码编译命令。当前版本只支持make。
目标操作系统	选择源码将要移植到的目标服务器操作系统型号和版本。
目标系统内核版本	选择目标操作系统内核版本。

The screenshot shows a web interface for configuring a source code analysis task. It includes the following fields and options:

- 源代码存放路径**: /opt/portadv/portadmin/ (with a text input field below it)
- 检查项目**: 编译器选项、编译器宏、汇编程序、内置函数、属性
- 编译器版本**: GCC 4.8 (dropdown menu)
- 构建工具**: make (dropdown menu)
- * 编译命令**: make (dropdown menu)
- 目标操作系统**: Centos7.6 (dropdown menu)
- 目标系统内核版本**: v4.14 (dropdown menu)
- 分析**: A button to start the analysis.



源码分析 (3)

- Step 7 分析完成后，在右侧“历史报告”区域单击具体执行时间点的报告，即可查看详细的移植分析结果。

历史报告

2019/09/05 14:08:57



2019/09/05 12:16:42



2019/09/05 12:16:21





移植建议 (1)

- Step 1 打开移植分析报告。

< | 2019/09/18 12:10:08

移植报告 移植建议

2019/09/18 12:10:08

源代码存放路径	/opt/portadv/portadmin/	编译器版本	GCC 4.8	构建工具	make
编译命令	make	目标操作系统	Centos7.6	目标系统内核版本	v4.14

分析结果

- ▶ 需要移植的依赖库SO文件 54
- ▶ 需要移植的源文件 53
- 需要移植的代码行数 C/C++和Makefile源代码：102行；汇编代码：697行

下载.csv报告 下载.html报告



移植建议 (2)

- Step 2 查看分析结果。

- 查看需要移植的依赖库SO文件。

根据源码包中的{SO文件名}列表，对比SO依赖库白名单，得到需要移植的SO库的详细信息。

NO.	Name	Description	Operation Suggestion	Operation
0	liblibkae.so	--	--	Download
1	liblibssl.so	--	--	Download
2	libkae.so	--	--	Download
3	libnsl.so	--	--	Download
4	libtcmmalloc.so	--	--	Download

- 查看需要移植的C/C++源文件。

扫描分析用户软件目标二进制文件依赖的源文件集合，根据编译器版本信息，检查源码中使用的架构相关的编译器宏、buildin函数、attribute函数、用户自定义宏等，确定需要移植的源码及源文件。

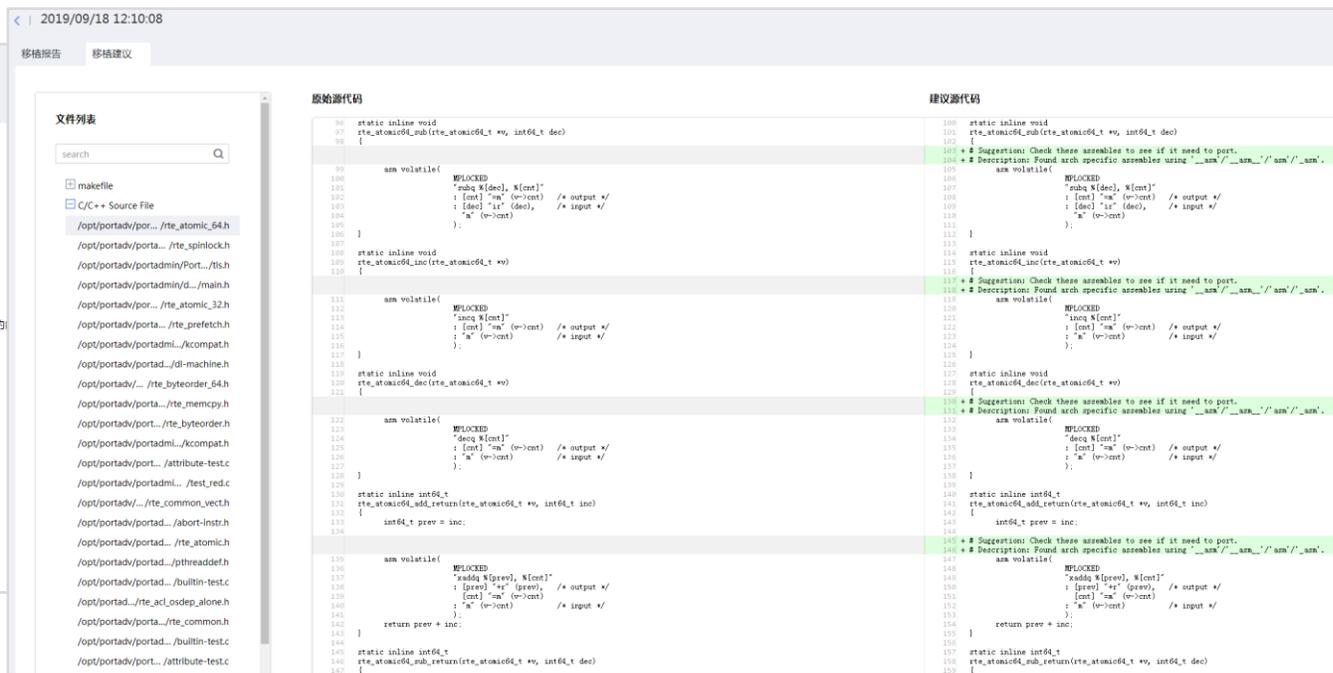
```
/opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/arch/x86/rte_byteorder.h /opt/portadv/portadmin/PortingTestData/simpleTest2/builtin-test.c
/opt/portadv/portadmin/PortingTestData/simpleTest/attribute-test.c /opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/arch/x86/rte_atomic_64.h
/opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/arch/x86/rte_spinlock.h /opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/arch/x86/rte_cpuflags.h
/opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/arch/x86/rte_cycles.h /opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/rte_common.h
/opt/portadv/portadmin/dpdk/lib/librte_eal/linuxapp/kni/ethtool/igb/kcompat.h /opt/portadv/portadmin/PortingTestData/asm/dl-machine.h
/opt/portadv/portadmin/dpdk/lib/librte_acl/rte_acl_osdep_alone.h /opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/arch/x86/rte_byteorder_64.h
/opt/portadv/portadmin/PortingTestData/asm/tls.h /opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/arch/x86/rte_atomic_32.h
/opt/portadv/portadmin/dpdk/lib/librte_eal/common/include/arch/x86/rte_memcpy.h /opt/portadv/portadmin/dpdk/lib/librte_eal/linuxapp/kni/ethtool/ixgbe/kcompat.h
```

- 查看需要移植的C/C++代码、汇编代码的总行数统计信息



移植建议 (3)

- Step 3 (可选) 您可以将分析报告下载到本地进行查阅。
 - 支持下载.csv格式和.html格式的分析报告。
- Step 4 单击“移植建议”页签。
- Step 5 阅读源码查看说明, 勾选“显示源代码”, 单击“确定”。
- Step 6 查看源码移植建议。





编译安装与测试验证

编译安装:

- Step 1 配置编译环境。
 - 1. 根据移植分析报告中对依赖库SO文件的操作建议，将依赖库SO文件下载并上传到TaiShan服务器或华为鲲鹏云服务器。
 - 2. 安装或编译依赖库SO文件。
- Step 2 修改源代码。
 - 1. 根据移植建议中的源码修改建议，修改源代码。
 - 2. 将修改后源代码文件上传至TaiShan服务器或华为鲲鹏云服务器。
- Step 3 编译安装。
 - 1. 执行编译。
 - 2. 执行安装。

应用程序编译安装后，需要通过一些测试工具、测试方法等验证应用是否能够正常运行在TaiShan服务器或华为鲲鹏云服务器上:

- Step 1 启动并运行应用程序。
- Step 2 验证应用程序的功能和性能。



CLI方式进行代码分析

- **命令功能**

对用户上传的源代码进行扫描分析，并输出详细的移植建议和指导。

- **命令格式**

`/opt/portadv/tools/cmd/bin/porting_advisor -S <source> -C <compiler> --cmd <cmd> --tos <tos> --tk <tk> -D <debug> -O <ouput>`

- **参数说明**

命令	参数	说明
-S/--source	source	必配参数，待扫描的C/C++源文件所在路径。 路径使用全路径。可以输入多个路径，用“,” 隔开。例如：/path/to/sourcecode
-C/--compiler	compiler	可选参数，指定GCC编译器版本，当前只支持gcc4.8。
-T/--tools	tools	可选参数，指定构建工具及命令行，make或者cmake，默认是make（当前版本只支持make）。
--cmd	cmd	必配参数，提供完整的软件构建命令，构建命令中必须要有make字段。例如：'make all'
--tos	tos	必配参数，软件需要移植到的目标服务器操作系统型号和版本。例如：CentOS7.6
--tk	tk	可选参数，软件需要移植到的目标操作系统内核版本。例如：4.14
-D/--debug	debug	可选参数，工具调试日志信息等级，分INFO、WARN、ERR三级。默认等级是INFO。 INFO log最多，ERR log最少。
-O/--ouput	ouput	可选参数，指定输出报告格式，csv或html格式（当前版本只支持csv）。例如：csv

- **使用实例**

以分析PortingTestData应用源代码并输出csv格式的分析报告为例，请根据实际情况将“/opt/code/PortingTestData/” 替换成实际需要扫描的代码路径。

`/opt/portadv/tools/cmd/bin/porting_advisor -S /opt/code/PortingTestData/ -C gcc4.8 --cmd 'make all' --tos centos7.6`



目录

1. 软件迁移原理和迁移过程
- 2. 迁移工具和迁移指导**
 - 华为鲲鹏代码迁移工具介绍
 - C/C++类应用移植
 - Java类应用移植
3. 容器迁移指导
4. 迁移常见问题及解决思路



Java类应用移植

- 基于解释型语言开发的应用程序，是CPU架构不相关的，例如Java、Python，将这类应用程序移植到TaiShan服务器或华为鲲鹏云服务器，无需修改和重新编译，按照与x86一致的方式部署和运行应用程序即可。
- Java应用程序jar包内，可能包含基于C/C++语言开发的SO库文件，这类SO库需要移植编译，使用编译得到的SO库重新打包jar包。移植具体方法与C/C++类应用相同。本节不再详细赘述。



思考题

1. 华为鲲鹏代码迁移工具适用于以下哪些类型的应用程序？（ ）
 - A. C/C++
 - B. Java
 - C. 汇编
 - D. Python
2. 华为鲲鹏代码迁移工具能够提供（ ）方面的移植评估结果。
 - A. 扫描源码中有多少个安装包
 - B. 扫描源码中有多少可以移植的依赖库SO文件
 - C. 扫描源码中有多少行可以移植的C/C++代码、汇编代码
 - D. 预估移植所需的工作量



本节小结

- 简要介绍代码迁移工具的定位、应用场景，支持的功能特性，以及用户如何访问和使用工具的方式。
- 介绍使用代码迁移工具如何对C/C++类应用进行移植的具体操作指导。
- 介绍需要进行移植的Java类应用程序场景说明。



目录

1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
- 3. 容器迁移指导**
4. 迁移常见问题及解决思路



本节概述和学习目标

- 上一节讲述了使用迁移工具迁移已有的源码到华为鲲鹏平台上，若华为鲲鹏平台的载体不是云服务器或物理服务器，而是基于鲲鹏平台开发的容器的时候如何进行容器的迁移呢？
- 学完本课程后，您将能够：
 - 能够理解容器和镜像的相关概念，容器与虚拟机的区别
 - 能够在鲲鹏服务器产品上通过Docker pull和Dockerfile方式构建基础镜像
 - 能够通过docker安装redis应用
 - 能够验证docker安装的redis应用是否成功



目录

1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
- 3. 容器迁移指导**
 - 容器相关概念介绍
 - 容器迁移背景和原理
 - 容器迁移主要流程
4. 迁移常见问题及解决思路



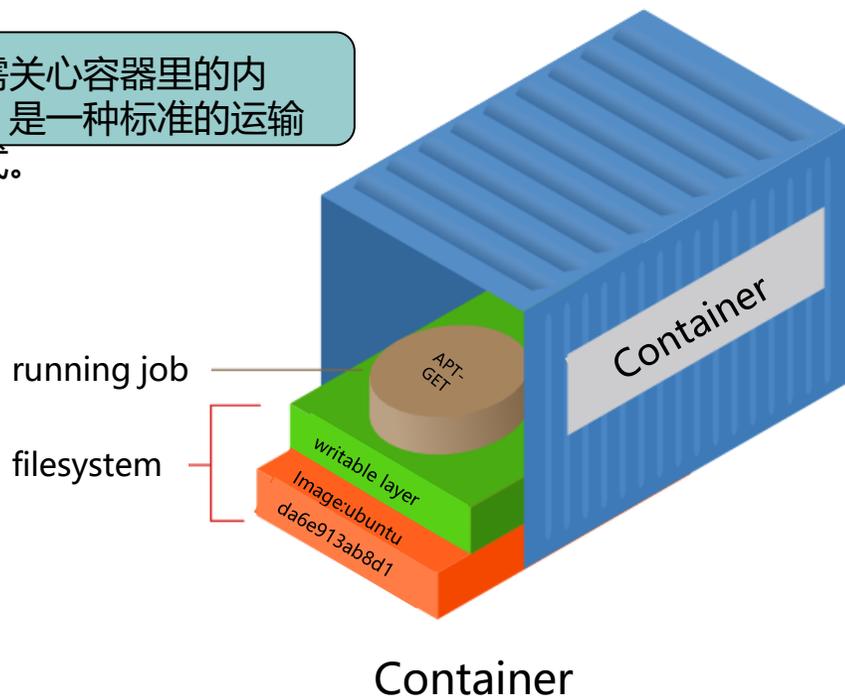
什么是容器

- 容器是一种轻量级、可移植、自包含的软件打包技术，使应用程序可以在几乎任何地方以相同的方式运行。开发人员在自己笔记本上创建并测试好的容器，无需任何修改就能够在生产系统的虚拟机、物理服务器或公有云主机上运行。



运输业

不需关心容器里的内容，是一种标准的运输方式。



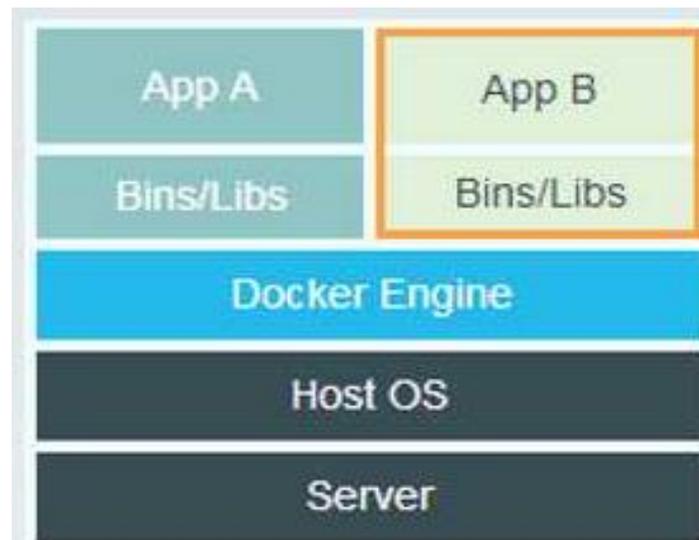


容器与虚拟机

- 容器和虚拟机之间的主要区别在于虚拟化层的位置和操作系统资源的使用方式。



虚拟机

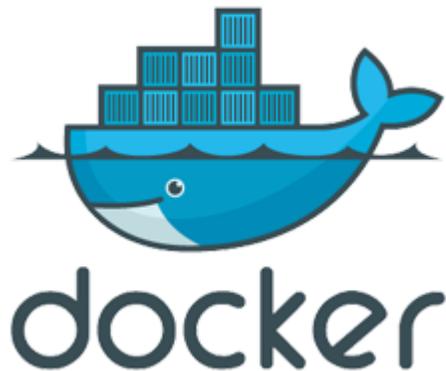


容器



什么是Docker

- Docker 是最受大众关注的容器技术，并且现在“几乎”成为事实上的容器标准。
- 简单的应用场景
 - Web 应用的自动化打包和发布。
 - 自动化测试和持续集成、发布。





容器与虚拟机参数对比

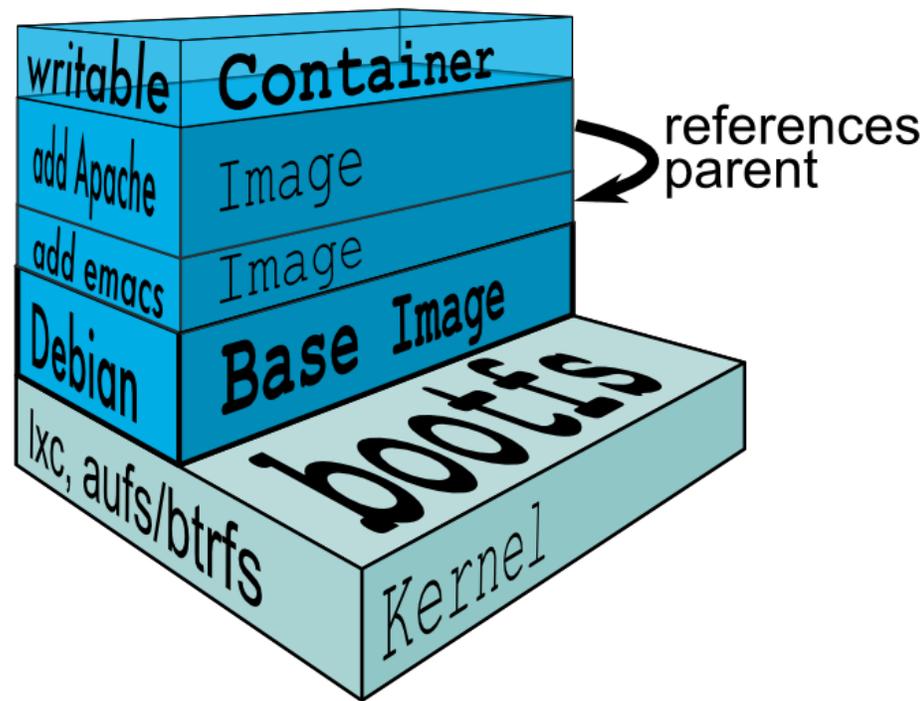
参数	容器 (Docker)	对比	虚拟机
快速创建, 删除	启动应用	>	启动Guest OS+启动应用
交互, 部署	容器镜像	=	虚拟机镜像
密度	单Node 100~1000	>	单Node 10~100
更新管理	迭代式更新, 通过修改Dockerfile, 对增量内容进行分发, 存储, 节点启动	>	向虚拟机推送安装, 升级应用软件补丁包
启动时间	秒级启动	>	分钟级启动
轻量级	镜像大小通常以M为单位	>	虚拟机以G为单位
性能	docker共享宿主机内核, 系统级虚拟化, 占用资源少, 没有Hypervisor层开销, 性能基本接近物理机	>	虚拟机需要Hypervisor层支持, 虚拟化一些设备, 具有完整的GuestOS, 虚拟化开销大, 因而降低性能, 没有容器性能好。
安全性	Docker具有宿主机root权限, 有一定安全隐患	<	硬件隔离, 相对安全
高可用性	通过业务本身的高可用性来保证	<	武器库丰富: 快照, 克隆, HA, 动态迁移, 异地容灾, 异地双活
使用要求	共享宿主机内核, 不用考虑CPU是否支持虚拟化技术	>	基于硬件的完全虚拟化, 需要硬件CPU虚拟化技术支持



Docker容器与镜像

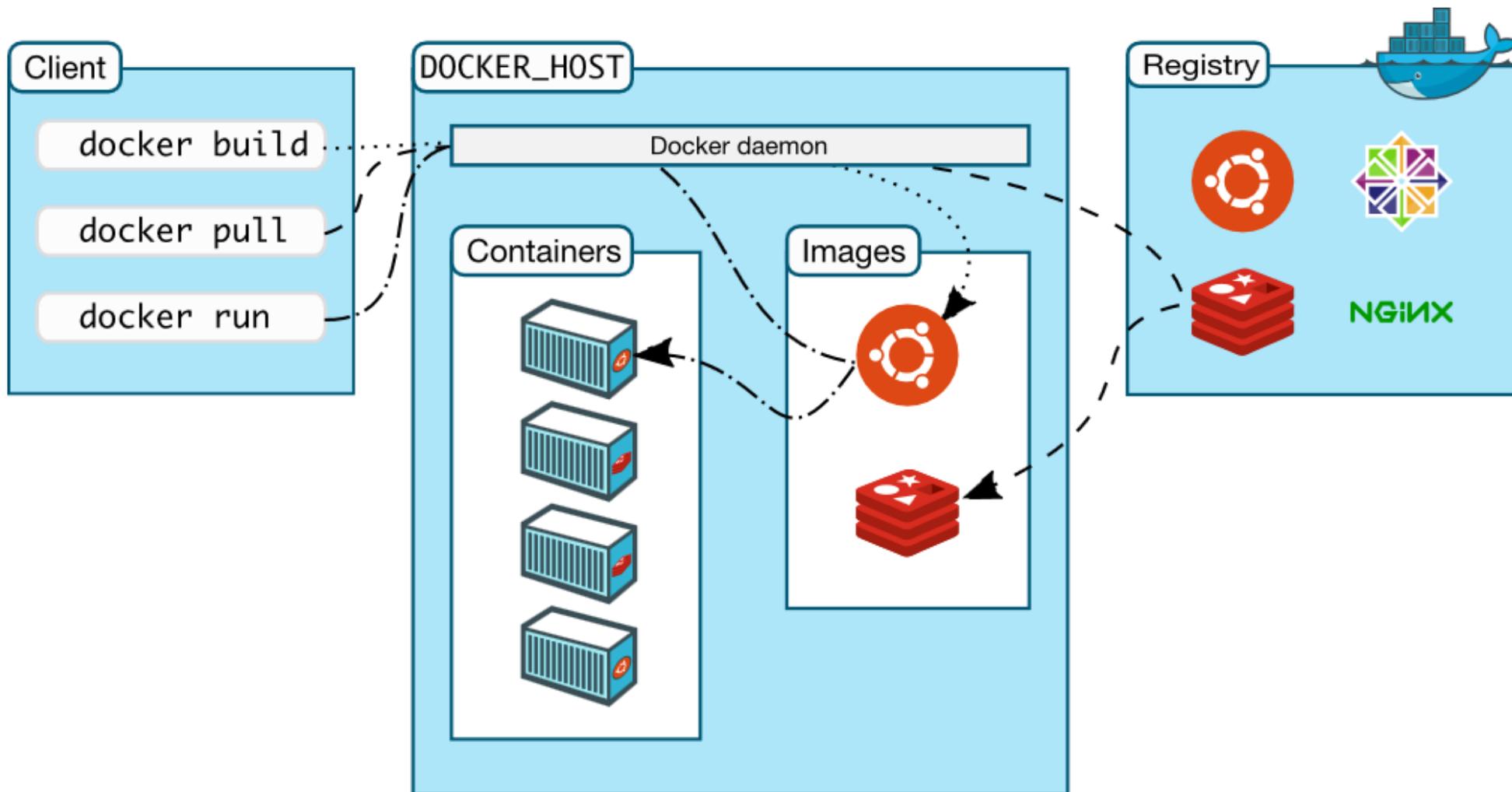
- Docker容器通过 Docker 镜像来创建
- 容器与镜像的关系类似于面向对象编程中的对象与类

Docker	面向对象
容器	对象
镜像	类





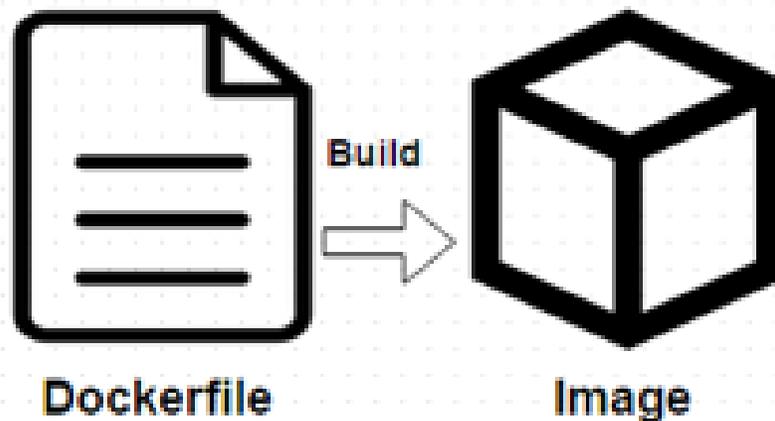
Docker架构





Dockerfile

- **Dockerfile**是一个文本格式的配置文件，包含创建镜像所需要的全部指令。基于在Dockerfile中的指令，用户可以快速创建自定义的镜像。
 - Docker image的表述
 - 简单语法用来构建镜像
 - 自动化的脚本创建镜像





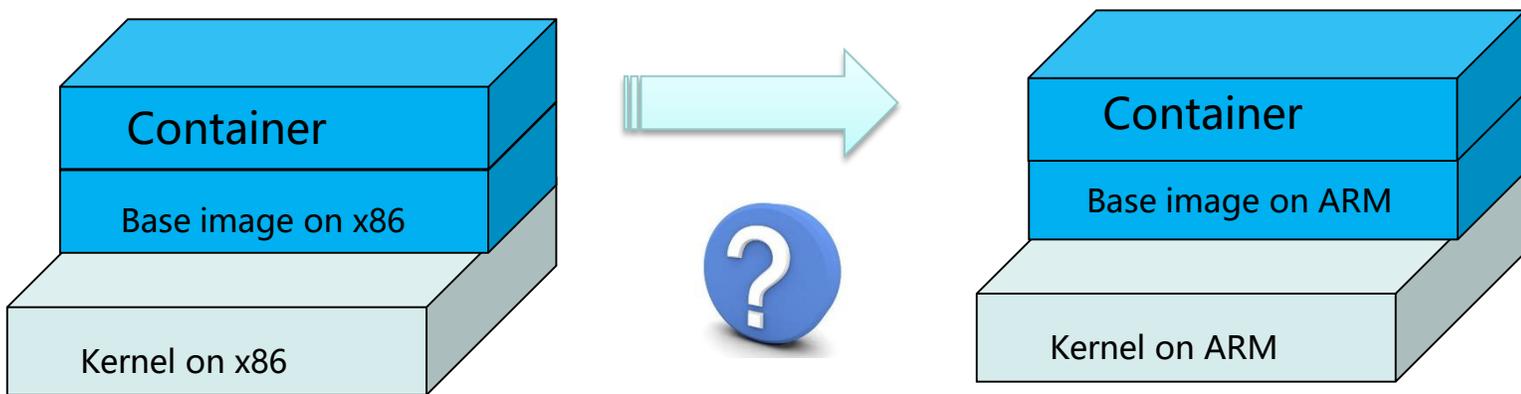
目录

1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
- 3. 容器迁移指导**
 - 容器相关概念介绍
 - 容器迁移背景和原理
 - 容器迁移主要流程
4. 迁移常见问题及解决思路



容器迁移的背景

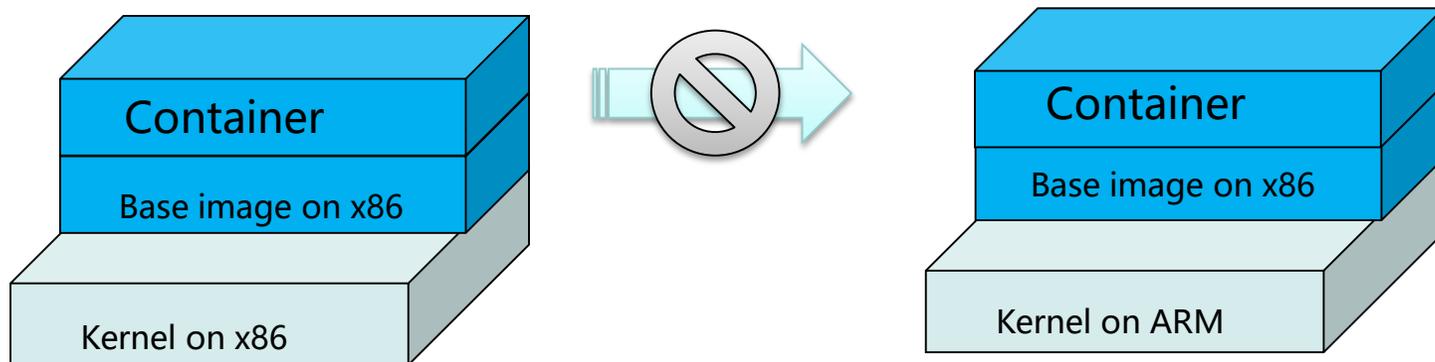
- 随着Docker技术应用领域的不断拓展， Docker容器使用所遇到的问题也显露无遗，主要表现在：系统负载不均衡和主机硬件维护困难。
- 为了能够有效解决上述问题， 彰显Docker容器的优势， 研究人员提出了Docker容器迁移策略。





容器运行的前提

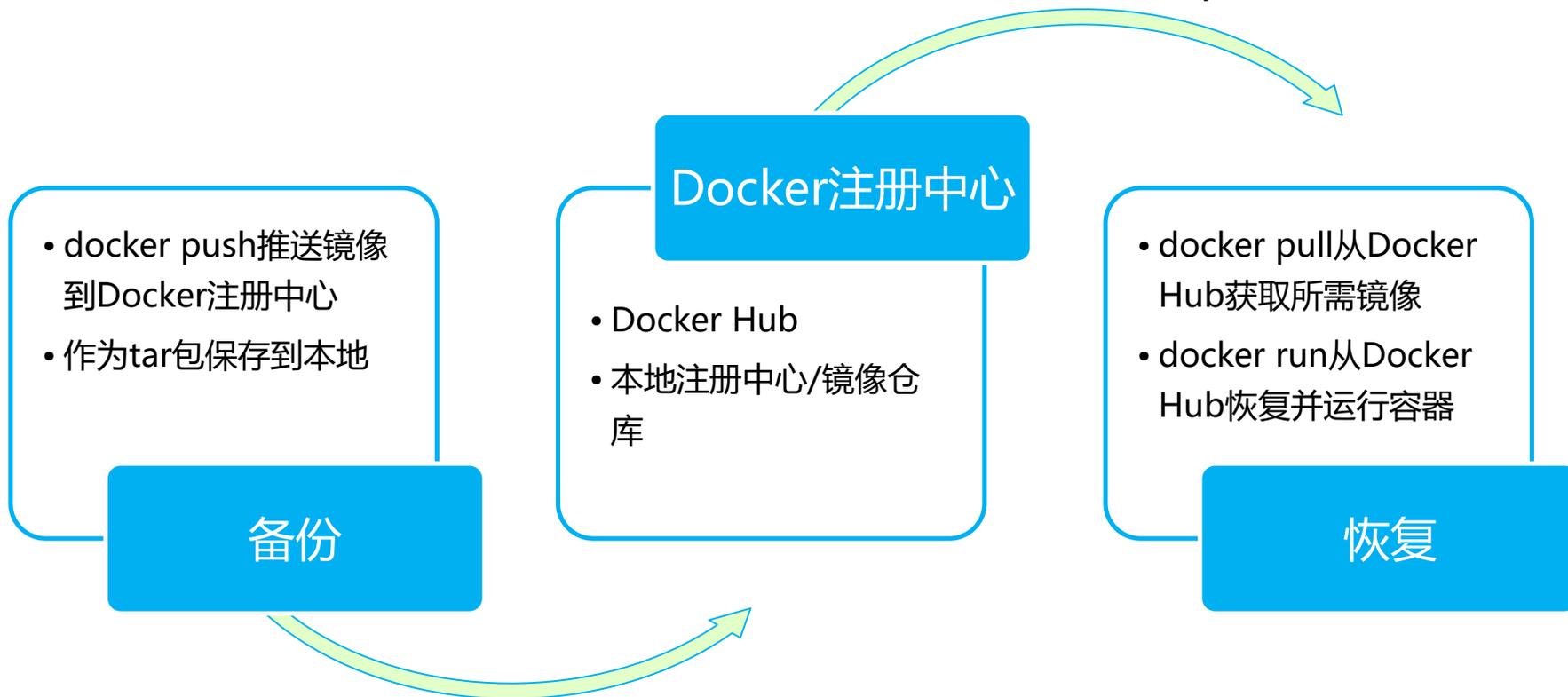
- 跨平台的容器无法运行，会出现格式错误
 - x86平台获取的镜像是适用于x86平台，当迁移到鲲鹏平台，容器无法执行。
 - 在基于ARM的平台中，docker pull方式或者Dockerfile方式获取或者构建的镜像均为基于ARM平台的，同样也无法在x86上运行。





容器迁移的原理

- 迁移容器同时涉及到了上面两个操作，备份和恢复。我们可以将任何一个Docker容器从一台机器迁移到另一台机器。在迁移过程中，首先我们将把容器备份为Docker镜像快照。然后，该Docker镜像或者是被推送到了Docker注册中心，或者被作为tar包文件保存到了本地。如果我们将镜像推送到了Docker注册中心，我们简单地从任何我们想要的机器上使用 `docker run` 命令来恢复并运行该容器，或者通过 `docker pull` 命令拉取我们想要的镜像。





Docker容器迁移策略

- Docker容器迁移有两种策略：使用Docker pull获取镜像或使用Dockerfile构建镜像。

Docker pull获取镜像

- Docker pull命令直接获取基于ARM平台的docker镜像
- 使用Docker pull命令的对象必须是适用于ARM的镜像，如arm64v8/centos:7

Dockerfile构建镜像

- 使用dockerfile构建基于ARM平台的docker镜像
- 基础镜像和构建后的镜像均只能使用于ARM平台



目录

1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
- 3. 容器迁移指导**
 - 容器相关概念介绍
 - 容器迁移背景和原理
 - 容器迁移主要流程
4. 迁移常见问题及解决思路



容器迁移整体步骤

• Docker安装

• Docker构建基础镜像

• Dockerfile创建应用镜像

• 验证应用镜像



检查环境是否支持Docker

- 检查内核版本

- `uname -r`

```
[root@worker01 ~]# uname -r  
4.14.0-115.5.1.el7a.aarch64
```

- Docker支持的CentOS版本要求

- CentOS 7, 系统为64位、系统内核版本为 3.10 以上

- CentOS 6.5或更高, 系统为64位、系统内核版本为 2.6.32-431 或者更高版本



安装和启动Docker

- 安装依赖软件包
- 安装 Docker
 - **yum -y install docker**
- 启动 Docker服务
 - **systemctl start docker**
- 详细参考<https://docs.docker.com/install/linux/docker-ce/centos/>



运行hello-world

- 运行 hello-world
 - 由于本地没有hello-world这个镜像，所以会下载一个hello-world的镜像，并在容器内运行
 - 对于本地没有的镜像，相当于先pull再run一个镜像
- docker run hello-world

```
[root@worker01 ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
3b4173355427: Pull complete
Digest: sha256:b8ba256769a0ac28dd126d584e0a2011cd2877f3f76e093a7ae560f2a5301c00
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (arm64v8)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

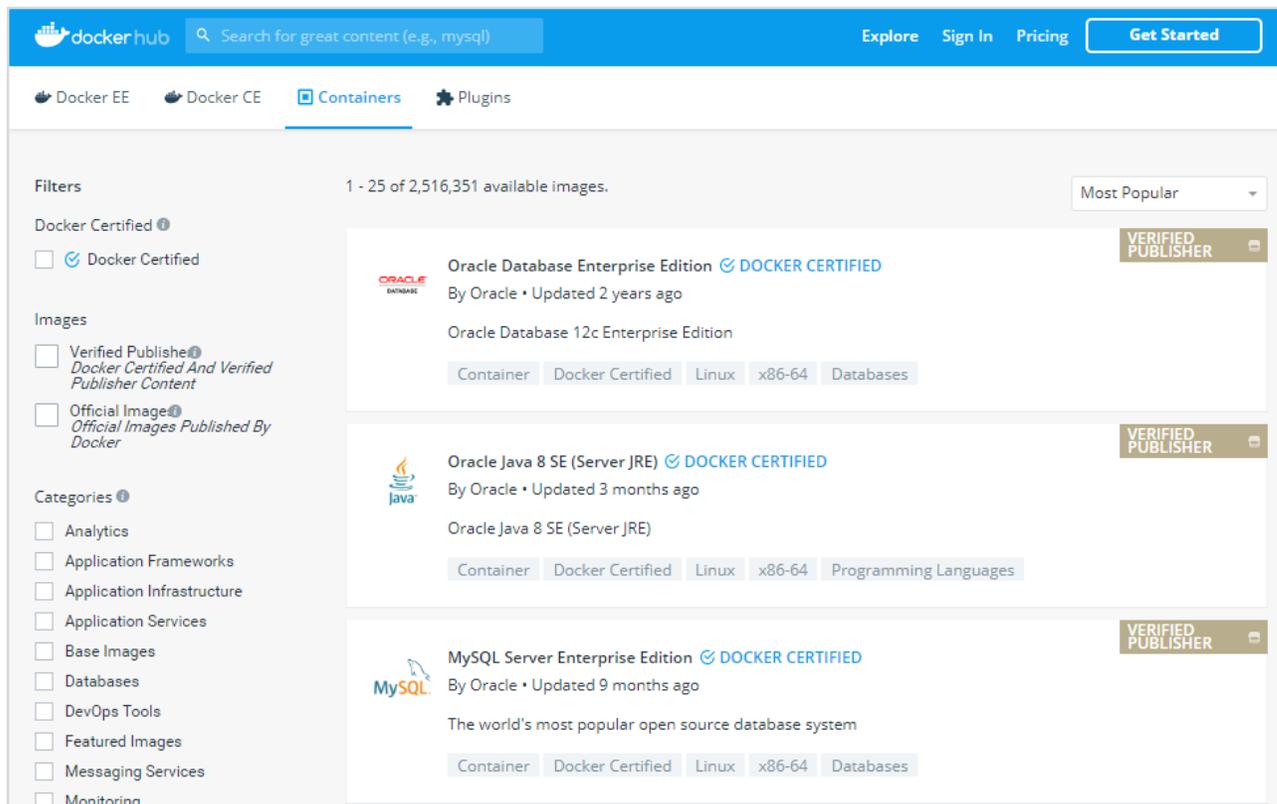
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```



Docker Hub

- Docker Hub是目前世界上最大的容器镜像仓库，由Docker公司维护。上面有Docker公司提供的镜像及大量用户上传的镜像。





查找镜像

- 查找Docker Hub上的CentOS镜像
 - docker search centos

```
[root@worker01 ~]# docker search centos
```

NAME	DESCRIPTION	STARS	OFFICIAL
centos	The official build of CentOS.	5566	[OK]
ansible/centos7-ansible	Ansible on Centos7	123	[OK]
jdeathe/centos-ssh	CentOS-6 6.10 x86_64 / CentOS-7 7.6.1810 x86...	112	[OK]
consol/centos-xfce-vnc	Centos container with "headless" VNC session...	99	[OK]
centos/mysql-57-centos7	MySQL 5.7 SQL database server	62	[OK]
imagine10255/centos6-lamp-php56	centos6-lamp-php56	57	[OK]
tutum/centos	Simple CentOS docker image with SSH access	45	[OK]
centos/postgresql-96-centos7	PostgreSQL is an advanced Object-Relational ...	39	[OK]



获取镜像

- 从docker hub拉取官方的镜像arm64v8/centos，标签为7
 - docker pull arm64v8/centos:7

```
[root@worker01 ~]# docker pull arm64v8/centos:7
7: Pulling from arm64v8/centos
4856e02b0d50: Pull complete
Digest: sha256:df89b0a0b42916b5b31b334fd52d3e396c226ad97dfe772848bdd6b00fb42bf0
Status: Downloaded newer image for arm64v8/centos:7
docker.io/arm64v8/centos:7
```

- 查看拉取官方的镜像arm64v8/centos，本地镜像列表里查到REPOSITORY为arm64v8/centos，标签为7的镜像时，说明拉取成功
 - docker images arm64v8/centos:7

```
[root@worker01 ~]# docker images arm64v8/centos:7
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
arm64v8/centos      7            0cb4fb73950e     4 weeks ago     239MB
```



启动和查看容器

- 使用镜像arm64v8/centos:7以交互模式启动一个容器，在容器内执行/bin/bash命令

- docker run -it arm64v8/centos:7 /bin/bash

```
[root@worker01 ~]# docker run -it arm64v8/centos:7 /bin/bash
[root@f4603f1f918a /]#
```

- 如果要退出就按Ctrl-D，或者输入exit
- 查看容器运行中和未运行的容器

- docker ps -a

```
[root@worker01 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
NAMES
f4603f1f918a       arm64v8/centos:7  "/bin/bash"        14 hours ago       Exited (0) 2 minutes ago
busy_khorana
7c69973c7050       hello-world        "/hello"           14 hours ago       Exited (0) 14 hours ago
mystifying_brown
```



提交和查看新镜像

- 根据容器ID创建一个新的镜像作为Redis的基础镜像
 - `docker commit -a "huawei.com" -m "redis images" f4603f1f918a t_arm64v8/centos:7`

```
[root@worker01 ~]# docker commit -a "huawei.com" -m "redis images" f4603f1f918a t_arm64v8/centos:7
sha256:ac4b911562b0b8ec231cb4a4e216f17b524f90e088742f557c110de7456f0c25
[root@worker01 ~]#
```

- 查看新构建的基础镜像
 - `docker images`

```
[root@worker01 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
t_arm64v8/centos    7                   ac4b911562b0       48 seconds ago     385MB
arm64v8/centos      7                   0cb4fb73950e       4 weeks ago        239MB
hello-world         latest              de6f0c40d4e5       8 months ago       4.75kB
```



创建redis目录

- 创建redis目录
 - `mkdir -p ~/redis ~/redis/data`
 - data目录将映射为redis容器配置的/data目录，作为redis数据持久化的存储目录
- 创建redis容器时候会自动产生一些数据，为了不让数据随着container的消失而消失，保证数据的安全性。例如：数据库容器，数据表的表会产生一些数据，如果把container给删除，数据就丢失。为了保证数据不丢失，这就有了Volume的存在。



Dockerfile编写

- 进入创建的redis目录，创建Dockerfile
 - Dockerfile编写内容如下：

```
FROM t_arm64v8/centos:7
WORKDIR /home
RUN wget
http://download.redis.io/releases/redis-
5.0.5.tar.gz && \
tar -xvzf redis-5.0.5.tar.gz && \
mv redis-5.0.5/ redis && \
rm -f redis-5.0.5.tar.gz
WORKDIR /home/redis
RUN make && make install
VOLUME /data

EXPOSE 6379
CMD ["redis-server"]
```

- Dockerfile参数内容如下：

FROM <image>或者FROM <image>:<tag>：基于哪个镜像，<image> 首选本地是否存在，如果不存在则会从公共仓库下载

WORKDIR：切换目录用，可以多次切换（相当于cd命令），对RUN, CMD, ENTRYPOINT生效

RUN：执行指令

EXPOSE：EXPOSE指令告诉容器在运行时要监听的端口，但是这个端口是用于多个容器之间通信用的（links），外面的host是访问不到的。要把端口暴露给外面的主机，在启动容器时使用-p选项

VOLUME：指定挂载点/data

CMD：用于容器启动时指定的服务



通过Dockerfile创建镜像

- 通过Dockerfile创建redis镜像
 - `docker build -t t_arm64v8/centos_redis:5.05 .`

```
[root@worker01 redis]# docker build -t t_arm64v8/centos_redis:5.05 .
Sending build context to Docker daemon 124.6MB
Step 1/7 : FROM t_arm64v8/centos:7
--> ac4b911562b0
Step 2/7 : WORKDIR /home
--> Running in 27e0d1db3f7f
Removing intermediate container 27e0d1db3f7f
--> 6d947ee18245
Step 3/7 : RUN wget http://download.redis.io/releases/redis-5.0.5.tar.gz && tar -xvzf redis-5.0.5.tar.gz && mv redis-5.0.5/ redis && rm -f redis-5.0.5.tar.gz
--> Running in c4380adfd537
--2019-09-21 02:58:18-- http://download.redis.io/releases/redis-5.0.5.tar.gz
Resolving download.redis.io (download.redis.io)... 109.74.203.151
Connecting to download.redis.io (download.redis.io)|109.74.203.151|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1975750 (1.9M) [application/x-gzip]
Saving to: 'redis-5.0.5.tar.gz'
```



执行redis-server

- 运行容器，执行redis-server
 - `docker run -p 6379:6379 -v $PWD/data:/data -d t_arm64v8/centos_redis:5.05 redis-server --appendonly yes`

```
[root@worker01 redis]# docker run -p 6379:6379 -v $PWD/data:/data -d t_arm64v8/centos_redis:5.05 redis-server --appendonly yes
566f60cc5f6fedc3ac7c92c09ea36d79c1a9880459141dec492eaac3c3637ddf
```

- 命令说明：
 - `-p 6379:6379` : 将容器的6379端口映射到主机的6379端口
 - `-v $PWD/data:/data` : 将主机中当前目录下的data挂载到容器的/data
 - `redis-server --appendonly yes` : 在容器执行redis-server启动命令，并打开redis持久化配置



连接redis容器

- 执行redis-cli命令连接到刚启动的容器，即连接容器中的redis服务端
 - `docker exec -it 566f60cc5f6f redis-cli`
 - `127.0.0.1:6379> info`

```
[root@worker01 redis]# docker exec -it 566f60cc5f6f redis-cli
127.0.0.1:6379> info
# Server
redis_version:5.0.5
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:d66412af2962c379
redis_mode:standalone
os:Linux 4.14.0-115.5.1.el7a.aarch64 aarch64
arch_bits:64
multiplexing_api:epoll
atomicvar_api:atomic-builtin
gcc_version:4.8.5
process_id:1
run_id:db158c55dcb89559465f6f823cc7f054323b9d8a
tcp_port:6379
uptime_in_seconds:252
uptime_in_days:0
hz:10
configured_hz:10
lru_clock:8757623
executable:/home/redis/redis-server
config_file:
```



使用redis容器

- 连接容器中的redis服务端后，在交互命令中输入命令，即可使用redis了

```
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> set runkey "hello redis"
OK
127.0.0.1:6379> get runkey
"hello redis"
127.0.0.1:6379> █
```

- 说明
 - ping返回PONG说明检测到 redis 服务已经启动
 - set runkey "hello redis": 设置runkey值为"hello redis", 返回OK, 说明设置成功
 - get runkey: 获取runkey的值, 返回"hello redis"说明与设置的相匹配



思考题

1. 关于执行命令“docker ps -a”后，显示的标题含义描述，正确的是？（）
 - A. CONTAINER ID:容器的唯一表示ID
 - B. IMAGE:创建容器时使用的镜像
 - C. COMMAND:容器最后运行的命令
 - D. CREATED:创建容器的时间
2. 关于Docker的镜像仓库，说法正确的是？（）
 - A.实现Docker镜像的全局存储
 - B. 提供API接口
 - C.提供Docker镜像的下载/推送/查询
 - D.可用于租户管理



本节小结

- 本节主要讲述了什么是容器，以及在鲲鹏平台上如何进行容器的迁移，包括如下：
 - 容器的相关概念介绍
 - Docker容器迁移策略
 - Docker安装
 - Docker构建基础镜像
 - 根据基础镜像安装Redis
 - 验证Redis镜像



目录

1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
3. 容器迁移指导
- 4. 迁移常见问题及解决思路**



本节概述和学习目标

- 从前面章节我们了解了迁移原理和过程，以及在不同载体上的迁移流程和操作步骤，最后我们来了解下迁移过程中常见的问题
- 学完本节后，您将能够：
 - 熟悉软件迁移过程中常见的问题
 - 了解常见问题的解决思路



目录

1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
3. 容器迁移指导
- 4. 迁移常见问题及解决思路**
 - 常见编译问题
 - 常见功能问题
 - 常见工具问题



C/C++语言char数据类型默认符号不一致问题 (1)

问题现象

C/C++代码在编译时遇到如下提示:

告警信息: warning: comparison is always false due to limited range of data type

原因分析

char变量在不同CPU架构下默认符号不一致, 在x86架构下为signed char, 在ARM64平台为unsigned char, 移植时需要指定char变量为signed char。

解决方案

1. 在编译选项中加入“-fsigned-char”选项, 指定ARM64平台下的char为有符号数;
2. 将char类型直接声明为有符号char类型: signed char。



C/C++语言char数据类型默认符号不一致问题 (2)

鲲鹏弹性云服务器

```
[root@ecs-8225-kunpeng ~]#  
[root@ecs-8225-kunpeng ~]# uname -a  
Linux ecs-8225-kunpeng 4.14.0-115.5.1.el7a.aarch64 #1 SMP Mon Feb 4 16:38:08 UTC 2019 aarch64  
[root@ecs-8225-kunpeng ~]# cat charTest.c  
#include <stdio.h>  
int main()  
{  
    char ch=-1;  
    printf("ch=%d\n",ch);  
    return 0;  
}  
[root@ecs-8225-kunpeng ~]# gcc -o charTest charTest.c  
[root@ecs-8225-kunpeng ~]# ./charTest  
ch=255  
[root@ecs-8225-kunpeng ~]#
```

X86弹性云服务器

```
[root@ecs-82e5-x86 ~]# uname -a  
Linux ecs-82e5-x86 3.10.0-1062.1.1.el7.x86_64 #1 SMP Fri Sep 13 22:55:44 UTC 2019 x86_64  
[root@ecs-82e5-x86 ~]# cat charTest.c  
#include <stdio.h>  
int main()  
{  
    char ch = -1;  
    printf("ch=%d\n",ch);  
    return 0;  
}  
[root@ecs-82e5-x86 ~]# gcc -o charTest charTest.c  
[root@ecs-82e5-x86 ~]# ./charTest  
ch=-1  
[root@ecs-82e5-x86 ~]#
```

可以看到：相同的代码，鲲鹏下char默认为unsigned char类型，所以赋值为-1的时候，输出的为-1对256取模的结果即255，X86中的char默认为signed char类型，输出为-1



C/C++ 语言中调用汇编指令编译错误

问题现象

C/C++代码在编译时遇到如下提示：

错误信息：error: impossible constraint in 'asm' __asm__ __volatile__

原因分析

代码中使用汇编指令，而汇编指令与cpu指令集强相关。在x86架构cpu中的汇编指令需要修改为鲲鹏处理器平台的指令才能编译通过，实现功能替换。

解决方案

1. 本例中的代码调用了x86平台的汇编指令，修改为鲲鹏处理器对应的指令即可；
2. 部分功能可以修改为使用编译器自带的builtin函数，在基本不降低性能的前提下，提升代码的可移植性。



编译错误：无法识别-m64编译选项

问题现象

C/C++代码在编译时遇到如下提示：

错误信息：gcc: error: unrecognized command line option '-m64'

原因分析

-m64是x86 64位应用编译选项，m64选项设置int为32bits及long、指针为64 bits，为AMD的x86 64架构生成代码。在鲲鹏处理器平台无法支持。

解决方案

将鲲鹏处理器平台对应的编译选项设置为-mabi=lp64，重新编译即可。



目录

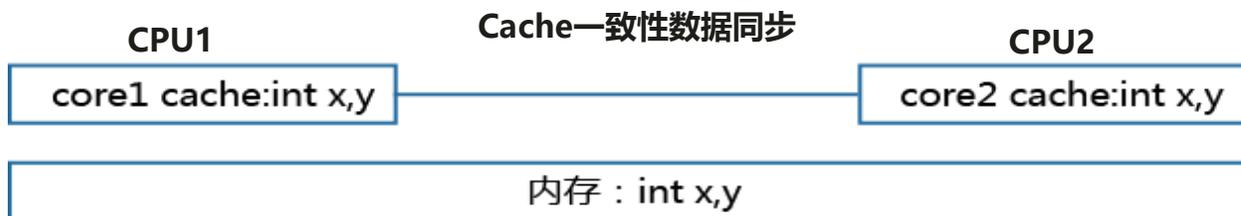
1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
3. 容器迁移指导
- 4. 迁移常见问题及解决思路**
 - 常见编译问题
 - 常见功能问题
 - 常见工具问题



弱内存序问题

弱内存序导致多线程程序执行结果概率性不一致

- 同一份数据，在cache里面存在多份，需要CPU之间进行同步



- 代码编写顺序和执行顺序可能不一样

开发人员写如下代码

```
int x = 0;
int y = 0;
...
x = 1; //x, y为全局变
y = 1; 量
```

CPU1上可能的执行顺序与预期不一致:

```
y = 1;
x = 1;
```

//x在内存中，等待导入，y在cache中，先执行

CPU2上的线程在执行如下逻辑，就可能出现问题:

```
if (y == 1){
    assert(x == 1)
}
```

//通过cache一致性同步y=1, 此时x=0

- CPU1: 执行x=1时，如果x在内存，y在cache，则core1等待x导入到cache，先执行y=1并同步给CPU2的 core2 cache
- CPU2: y=1成立，执行assert(x==1)错误 (assert是一种断言操作)

解决方案

- 使用内存屏障指令保证对共享数据的访问和预期一致

smp_wmb(): 保证写操作有序

smp_rmb(): 保证读操作有序

```
int x = 0;
int y = 0;
x = 1;
smp_wmb(); // 等待 x=1 执行完成
y = 1;
```

其它线程在执行如下逻辑，就可能出现问题

```
if (y == 1) {
    smp_rmb(); // 保证读的数据是最新的
    assert(x == 1);
}
```



超出整型取值范围时浮点型转整型与x86不一致

问题现象

C/C++双精度浮点型数转整型数据时，如果超出了整型的取值范围，鲲鹏处理器的表现与x86平台的表现不同。测试代码如下：

```
long aa = (long)0x7FFFFFFFFFFFFFFF;
long bb;
bb = (long)(aa*(double)10); // long->double->long
// x86 : aa=9223372036854775807, bb=-9223372036854775808
// Kunpeng : aa=9223372036854775807, bb=9223372036854775807
```

原因分析

x86（指令集）中的浮点到整型的转换指令，定义了一个indefinite integer value——“不确定数值”（64bit: 0x8000000000000000），大多数情况下x86平台确实都在遵循这个原则，但是在从double向无符号整型转换时，又出现了不同的结果。鲲鹏的处理则非常清晰和简单，在上溢出或下溢出时，保留整型能表示的最大值或最小值。

C/C++语言double类型超出整型取值范围向整型转换参照

CPU	double值	转为long变量保留值	CPU	double值	转为long变量保留值
x86	正值超出long范围	0x8000000000000000 0	x86	正值超出long范围	0x8000000000000000 00
x86	负值超出long范围	0x8000000000000000 0	x86	负值超出long范围	0x8000000000000000 00
鲲鹏	正值超出long范围	0x7FFFFFFFFFFFFFFF	鲲鹏	正值超出long范围	0x7FFFFFFFFFFFFFFF F
鲲鹏	负值超出long范围	0x8000000000000000 0	鲲鹏	负值超出long范围	0x8000000000000000 00

CPU	double值	转为long变量保留值	CPU	double值	转为long变量保留值
x86	正值超出long范围	0x8000000000000000 0	x86	正值超出long范围	0x8000000000000000 00
x86	负值超出long范围	0x8000000000000000 0	x86	负值超出long范围	0x8000000000000000 00
鲲鹏	正值超出long范围	0x7FFFFFFFFFFFFFFF	鲲鹏	正值超出long范围	0x7FFFFFFFFFFFFFFF F
鲲鹏	负值超出long范围	0x8000000000000000 0	鲲鹏	负值超出long范围	0x8000000000000000 00



对结构体中的变量进行原子操作时程序异常

问题现象

程序调用原子操作函数对结构体中的变量进行原子操作，程序coredump，堆栈如下：

```
Program received signal SIGBUS, Bus error.
0x00000000040083c in main () at /root/test/src/main.c:19
19  __sync_add_and_fetch(&a.count, step);
(gdb) disassemble
Dump of assembler code for function main:
0x000000000400824 <+0>:  sub  sp, sp, #0x10
0x000000000400828 <+4>:  mov   x0, #0x1          // #1
0x00000000040082c <+8>:  str  x0, [sp, #8]
0x000000000400830 <+12>: adrp  x0, 0x420000 <_libc_start_main@got.plt>
0x000000000400834 <+16>: add  x0, x0, #0x31 //将变量的地址放入x0寄存器
0x000000000400838 <+20>: ldr  x1, [sp, #8] //指定ldxr取数据的长度
(此处为8字节)
=> 0x00000000040083c <+24>: ldxr x2, [x0] //ldxr从x0寄存器指向的内存地址中取值
0x000000000400840 <+28>:  add  x2, x2, x1
0x000000000400844 <+32>:  stlxx w3, x2, [x0]
0x000000000400848 <+36>:  cbnz w3, 0x40083c <main+24>
0x00000000040084c <+40>:  dmb  ish
0x000000000400850 <+44>:  mov  w0, #0x0          // #0
0x000000000400854 <+48>:  add  sp, sp, #0x10
0x000000000400858 <+52>:  ret
End of assembler dump.
(gdb) p /x $x0
$4 = 0x420039 // x0寄存器存放的变量地址不在8字节地址对齐处
```

原因分析

鲲鹏处理器对变量的原子操作、锁操作等用到了ldaxr、stlxx等指令，这些指令要求变量地址必须按变量长度对齐，否则执行指令会触发异常，导致程序coredump。

一般是因为代码中对结构体进行强制字节对齐，导致变量地址不在对齐位置上，对这些变量进行原子操作、锁操作等会触发问题。

解决方案

代码中搜索“#pragma pack”关键字（该宏改变了编译器默认的对齐方式），找到使用了字节对齐的结构体，如果结构体中变量会被作为原子操作、自旋锁、互斥锁、信号量、读写锁的输入参数，则需要修改代码保证这些变量按变量长度对齐。



加速器初始化失败

问题现象

使用KAE引擎的测试RSA性能，性能与不启用加速引擎相当。

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae rsa2048
....
rsa 2048 bits 0.001359s 0.000043s 736.0 23458.9
```

原因分析

加速器未加载成功，需要检查加速器驱动、加速器引擎库相关的软链接、环境变量是否正确安装或配置。

解决方案

步骤1 检查如下加速器驱动是否加载成功加载到内核：uacce.ko、qm.ko、sgl.ko、hisi_sec2.ko、hisi_hpre.ko、hisi_zip.ko、hisi_rde.ko

```
[root@localhost ~]# lsmod | grep uacce
uacce 262144 2 hisi_hpre,qm
```

步骤2 检查/usr/lib64和OpenSSL安装目录是否有加速器引擎库，且建立正确的软链接

#查询kae是否正确安装并建立软连接，如果有正确安装显示如下内容

```
[root@localhost home]# ll /usr/local/lib/engines-1.1/ |grep kae
lrwxrwxrwx. 1 root root 22 Nov 12 02:33 kae.so -> kae.so.1.0.1
lrwxrwxrwx. 1 root root 22 Nov 12 02:33 kae.so.0 -> kae.so.1.0.1
-rwxr-xr-x. 1 root root 112632 May 25 2019 kae.so.1.0.1
```

```
[root@localhost home]# ll /usr/lib64/ | grep libwd
```

#查询wd是否正确安装并建立软连接，如果有正确安装显示如下内容

```
lrwxrwxrwx. 1 root root 14 Nov 12 02:33 libwd.so -> libwd.so.1.0.1
lrwxrwxrwx. 1 root root 14 Nov 12 02:33 libwd.so.0 -> libwd.so.1.0.1
-rwxr-xr-x. 1 root root 137120 May 25 2019 libwd.so.1.0.1
```

步骤3 检查环境变量LD_LIBRARY_PATH是否包含OpenSSL库路径，通过export命令增

```
[root@localhost home]# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
[root@localhost home]# echo $LD_LIBRARY_PATH
/usr/local/lib
```



目录

1. 软件迁移原理和迁移过程
2. 迁移工具和迁移指导
3. 容器迁移指导
- 4. 迁移常见问题及解决思路**
 - 常见编译问题
 - 常见功能问题
 - 常见工具问题



迁移工具源码路径错误

问题现象

使用Porting迁移工具执行分析任务时，填写源码存放路径后，执行分析任务，报“源代码路径错误”：

! 源代码路径错误。

鲲鹏代码迁移工具

分析软件源代码，提供应用移植指导。

源代码存放路径	/opt/portadv/portadmin/ /home/portingTest/
编译器版本	GCC 4.8
构建工具	make
* 编译命令	make
目标操作系统	Centos7.6

原因分析

源代码没有放置到porting工具的安装目录下

解决方案

在工具web界面，重新填写源代码存放路径，单击“分析”。弹窗页面显示任务分析进度，分析完成后，自动跳转至“移植报告”界面。

源代码存放路径	/opt/portadv/portadmin/ portingTest/
编译器版本	GCC 4.8
构建工具	make
* 编译命令	make
目标操作系统	Centos7.6
目标系统内核版本	v4.14

检查项目：编译器选项、编译器宏、汇编程序、内置函数、属性

分析



Maven软件仓库编译错误：未配置代理

问题现象

在通过代理访问网络的环境，如果没有配置代理编译会报错，如：

```
repo.maven.apache.org: Name or Service not known
```

原因分析

maven有自己的网络配置，如果服务器需要代理访问外网，则同时也需要给maven设置代理

解决方案

修改maven安装路径下的conf/settings.xml，在proxy标签中添加代理信息：

```
<proxies>
  <proxy>
    <id>optional</id>
    <active>true</active>
    <protocol>http</protocol>
    <username>zhangsan</username>
    <password>*****</password>
    <host>192.168.33.234</host>
    <port>8080</port>
    <nonProxyHosts>local.net|some.host.com</nonProxyHosts>
  </proxy>
</proxies>
```



Maven软件仓库编译错误：无法找到依赖库

问题现象

maven编译报错：

Failed to execute goal on project hadoop-auth: Could not resolve dependencies for project

org.apache.hadoop:hadoop-auth:jar:2.7.1.2.3.4.7-4: The following artifacts could not be resolved:

org.mortbay.jetty:jetty-util:jar:6.1.26.hwx,

org.mortbay.jetty:jetty:jar:6.1.26.hwx,

org.apache.zookeeper:zookeeper:jar:3.4.6.2.3.4.7-4:

Failure to find **org.mortbay.jetty:jetty-util:jar:6.1.26.hwx** in

<https://repository.apache.org/content/repositories/snapshots>

原因分析

从远程仓库下载jetty-util-6.1.26.hwx.jar包失败，原因是远程仓库中没有对应的jar。

解决方案

修改maven安装路径下的conf/settings.xml，在mirror标签中设置maven远程仓库路径，比如华为

```
<mirror>
  <id>mirror</id>
  <mirrorOf>*</mirrorOf>
  <name>huaweicloud</name>
  <url>https://repo.huaweicloud.com/repository/maven/</url>
</mirror>
```



思考题

1. 在向鲲鹏处理器迁移软件时，以下哪些是可能导致编译错误或告警的原因？（ ）
 - A. 编译选项
 - B. 数据类型不同
 - C. 汇编指令
 - D. 弱内存序问题
2. 弱内存序问题主要与如下那些因素相关？（ ）
 - A. 多线程
 - B. 多进程
 - C. 不同CPU之间Cache同步
 - D. 一级、二级、三级Cache间数据同步
 - E. 不同core之间Cache同步



本章总结

- 本章主要介绍了从应用从x86平台到鲲鹏平台需要进行软件迁移的背景、原理和实施流程，针对服务器和容器两种鲲鹏平台的应用，如何进行迁移，并阐述了迁移过程和主要操作。最后对迁移中所遇到的常见问题给出了解决思路。



学习推荐

- 《TaiShan代码移植指导》
 - <https://bbs.huaweicloud.com/forum/thread-22606-1-1.html>
- 华为云鲲鹏社区
 - <https://www.huaweicloud.com/kunpeng/>
- 《计算机组成与设计（ARM版）》

The image features a blue-tinted background with silhouettes of several groups of business professionals in a modern office environment. They are engaged in various activities like reviewing documents and talking. The overall aesthetic is professional and corporate.

谢谢

www.huawei.com



应用性能测试与调优



前言

- 本章主要介绍在数据库、大数据和HPC场景下进行性能测试的方法，使用华为鲲鹏性能优化工具Kunpeng Tuning Kit对C/C++类程序和Java Mixed-Mode类程序进行性能分析的方法，以及对NUMA和线程并发数进行性能调优的案例。



目标

- 学完本课程后，您将能够：
 - 掌握WRF在鲲鹏平台上进行性能测试的方法
 - 掌握使用BenchmarkSQL对数据库进行性能测试的方法
 - 掌握HiBench在鲲鹏平台上进行性能测试的方法
 - 掌握使用华为鲲鹏性能优化工具Kunpeng Tuning Kit进行性能分析的方法
 - 掌握对NUMA和线程并发数进行性能调优的方法



目录

1. 数据库性能测试
2. 大数据性能测试
3. HPC性能测试
4. 性能调优分析工具
5. 性能调优案例



本节概述和学习目标

- 本节主要介绍了如何使用开源数据库测试工具BenchmarkSQL对数据库进行性能测试的方法。
- 学完本节内容后，您将能够：
 - 了解TPCC测试模型及衡量标准
 - 掌握编译安装BenchmarkSQL的方法
 - 掌握执行BenchmarkSQL测试的方法



TPCC - 标准由来

- TPC (Transaction Processing Performance Council, 事务处理性能委员会) 是由数十家会员公司创建的非盈利组织, 总部设在美国。
- TPC的成员主要是计算机软硬件厂家, 其功能是制定商务应用基准程序的标准规范、性能和价格度量, 并管理测试结果的发布。
- TPC不给出基准程序的代码, 而只给出基准程序的标准规范。
- 任何厂家或其他测试者都可以根据规范, 最优地构造出自己的测试系统。
- 被测试者必须提交给TPC一套完整的报告, 包括被测系统的详细配置、分类价格等, 保证测试结果准确性。
- 测试模型包括TPC-C、TPC-H、TPC-DS、TPC-E等。

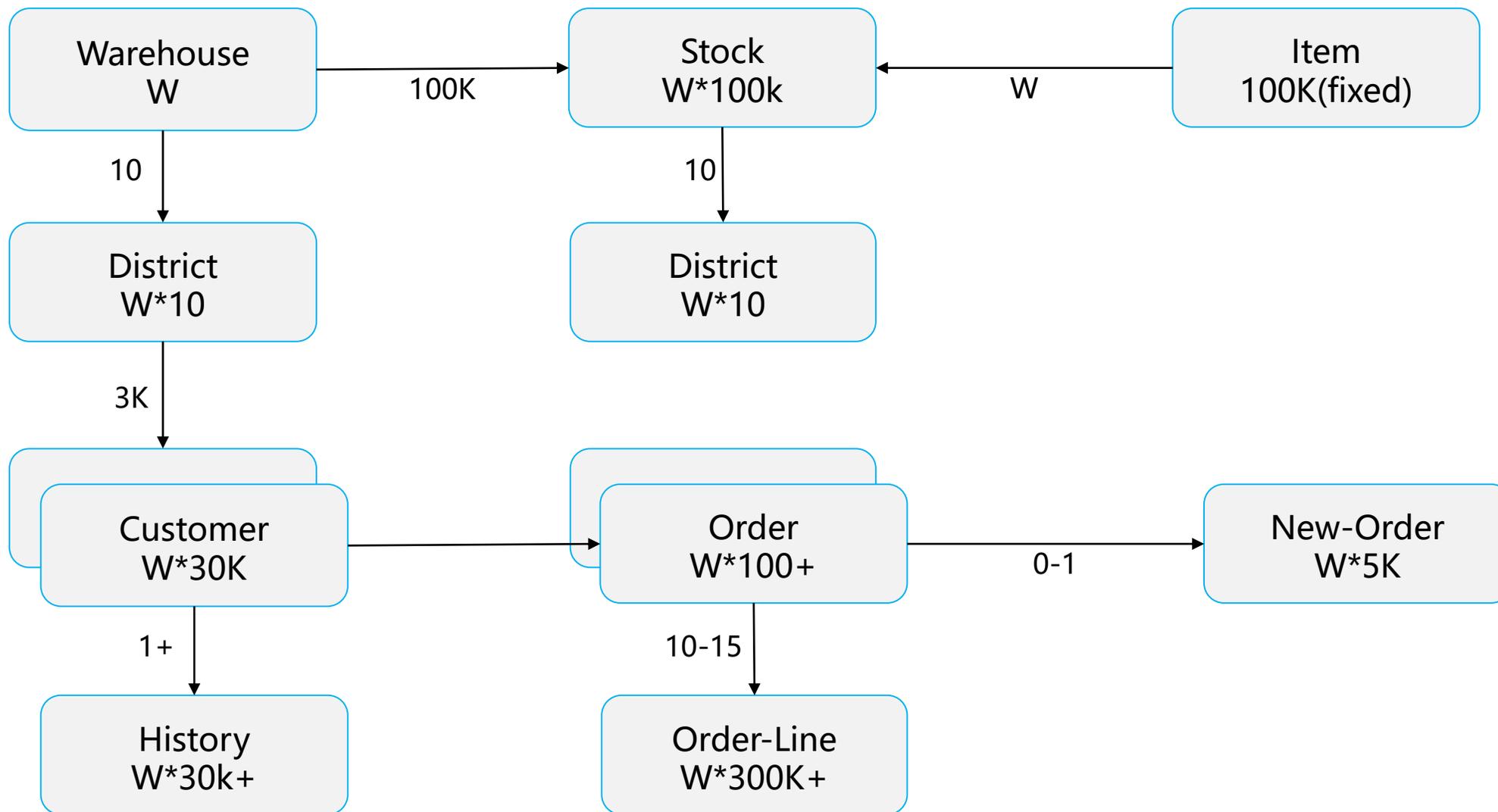


TPCC - 测试模型

- TPC-C是针对OLTP的基准测试模型，可以衡量数据库的性能和硬件性价比，是广泛应用并关注的一种测试模型。
- TPCC测试模型是一个大型的商品批发销售公司，它拥有若干个分布在不同区域的商品仓库。
 - 每个仓库负责为10个销售点供货，其中每个销售点为3000个客户提供服务。
 - 平均每个订单有10项产品。
 - 所有订单中约1%的产品在其直接所属的仓库中没有存货，必须由其他区域的仓库来供货。
 - 每个仓库都要维护公司销售的100000种商品的库存记录。



TPCC - 数据库模型





TPCC - 交易类型

- 新订单 (New-Order) : 客户输入一笔新的订货交易。
- 支付操作 (Payment) : 更新客户帐户余额以反映其支付状况。
- 发货 (Delivery) : 发货 (模拟批处理交易) 。
- 订单状态查询 (Order-Status) : 查询客户最近交易的状态。
- 库存状态查询 (Stock-Level) : 查询仓库库存状况, 以便能够及时补货。



TPCC - 性能衡量指标

- 流量指标
 - 系统在执行支付操作、订单状态查询、发货和库存状态查询这四种交易时，每分钟可以处理多少个新订单交易。
- 性价比
 - 测试系统的整体价格与流量指标的比值，在获得相同的流量指标即tpmC (transactions per minute, C即基准程序) 值的情况下，价格越低越好。



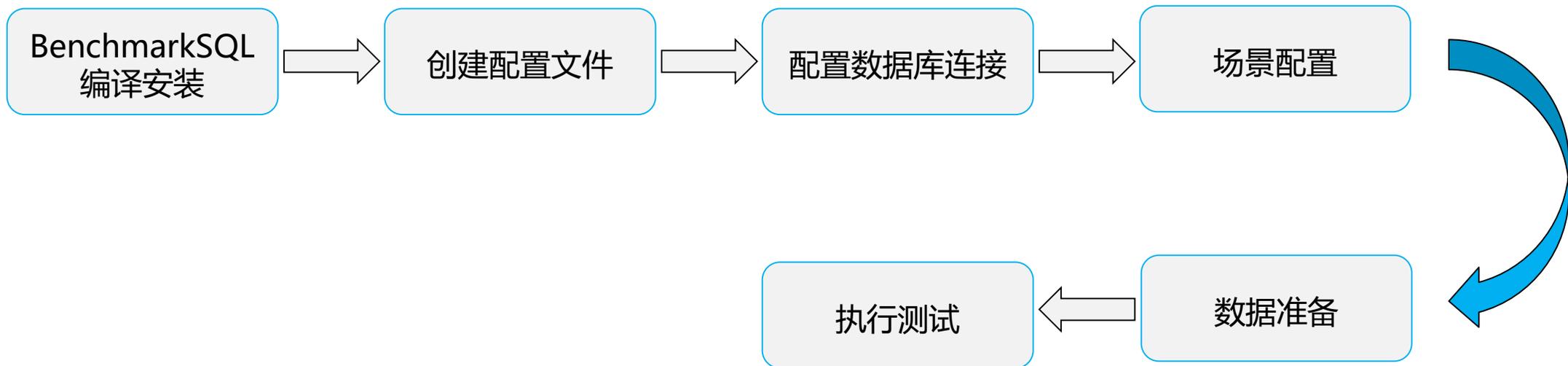
测试工具 - BenchmarkSQL

- Java语言编写，内嵌TPCC测试脚本的TPCC基础测试程序。
- 通过JDBC连接数据库，执行TPCC模型测试。
- 支持数据库：
 - Oracle
 - PostgreSQL
 - EnterpriseDB
 - DB2
 - SQL Server
 - GaussDB



测试流程

- 如下是使用测试工具BenchmarkSQL进行性能测试的流程图：





BenchmarkSQL - 编译安装

- 编译安装需求
 - JDK7或以上版本
- 编译安装
 - 解压缩源码包

```
Unzip benchmarksql-5.0.zip
```

- 切换到压缩包一级目录，执行ant编译

```
cd benchmarksql  
ant
```



BenchmarkSQL - 创建配置文件

- 配置文件
 - Bechmarksql/run/props.数据库类型, 例如:
 - props.ora: Oracle
 - props.pg: PostgreSQL
 - Props.fb: FirebirdSQL
 - Props.gb: GuassDB



BenchmarkSQL - 数据库配置

- 数据库连接
 - db: 数据库, 例如Oracle、PostgreSQL
 - driver: 数据库驱动
 - jdbc jar包: BenchmarkSQL\lib\数据库类型\
 - conn: 数据库连接字符串
 - user/password: 数据库用户名及密码



BenchmarkSQL - 场景配置

- warehouse: 指定仓库数量。
- loadWorkers: 指定装载数据的并发数。
- Terminals: 指定并发用户数。
- runMins: 指定测试时间。
- runTxnsPerTerminal: 指定每个Terminal运行的事务数量, runMins必须等于0。
- limitTxnsPerMin: 指定每分钟总事务数。
- terminalWarehouseFixed: 指定每个终端是否绑定固定warehouse。



BenchmarkSQL - 数据准备

- 数据加载

```
benchmarksql/run/runDatabaseBuild props.pg
```

- 创建数据库表
- 数据加载
- 创建索引



BenchmarkSQL - 执行测试

- 执行BenchmarkSQL测试

```
benchmarksql/run/runBenchmark.sh props.pg
```

- 测试结果

```
01:58:09,081 [Thread-1] INFO      jTPCC : Term-00,  
01:58:09,082 [Thread-1] INFO      jTPCC : Term-00, Measured tpmC (NewOrders) = 179.55  
01:58:09,082 [Thread-1] INFO      jTPCC : Term-00, Measured tpmTOTAL = 329.17  
01:58:09,082 [Thread-1] INFO      jTPCC : Term-00, Session Start      = 2016-05-25 01:58:07  
01:58:09,082 [Thread-1] INFO      jTPCC : Term-00, Session End        = 2016-05-25 01:58:09  
01:58:09,082 [Thread-1] INFO      jTPCC : Term-00, Transaction Count = 10
```

- 衡量标准

- tpmC (NewOrders)
- tpmTOTAL (TPS)



BenchmarkSQL - 性能优化思路

- BenchmarkSQL优化与测试数据库有关，比如MySQL数据库可以：
 - 优化文件系统
 - 使用xfs文件系统
 - 调整数据库参数
 - innodb_thread_concurrency #根据CPU核数进行调整
 - innodb_read_io_threads/innodb_write_io_threads #根据CPU核数和读写比例进行调整
 - innodb_buffer_pool_instances #根据资源争用情况进行调整
 - innodb_buffer_pool_size #根据系统内存进行调整
 - innodb_spin_wait_delay #根据系统压力及征用情况进行调整
 - table_open_cache #根据业务压力及状态变量Table_open_cache_misses进行调整



思考题

1. TPCC衡量标准是什么? ()
 - A. QphH
 - B. 响应时间
 - C. tpmC
 - D.TPS
2. BenchmarkSQL配置文件中loadWorkers指的是什么 () 。
 - A. 并发用户数
 - B. 数据库装载并发数
 - C. 数据库并行数
 - D. 数据库表的数量



本节小结

- 本节主要介绍了TPCC测试模型和衡量标准，介绍了编译安装数据库测试工具BenchmarkSQL，并使用BenchmarkSQL对数据库进行性能测试的方法。



目录

1. 数据库性能测试
2. 大数据性能测试
3. HPC性能测试
4. 性能调优分析工具
5. 性能调优案例



本节概述和学习目标

- 本节主要以HiBench为例，介绍了在大数据场景下进行性能测试的方法。
- 学完本节内容后，您将能够：
 - 理解大数据的基本概念
 - 掌握HiBench在鲲鹏平台上的性能测试方法和主要性能指标
 - 描述HiBench基本的性能优化思路



大数据简介

- 什么是大数据?
 - 大数据 (Big data) , 又称为**巨量资料**, 指的是在传统数据处理应用软件不足以处理的大或复杂的数据集的术语。
- 典型应用领域:

电商/零售领域



精准营销
预测趋势
生产管理
库存管理

金融领域



精准营销
风险管控
决策支持
效率提升
产品设计

医疗领域



数据分类
病情分析

交通领域



线路规划
效率提升

教育领域



效率评估
个性推送

环保领域



灾难预警
天气预报
污染分析
农业生产管理

舆情领域



犯罪管理
智能追踪



大数据基准测试套件：HiBench

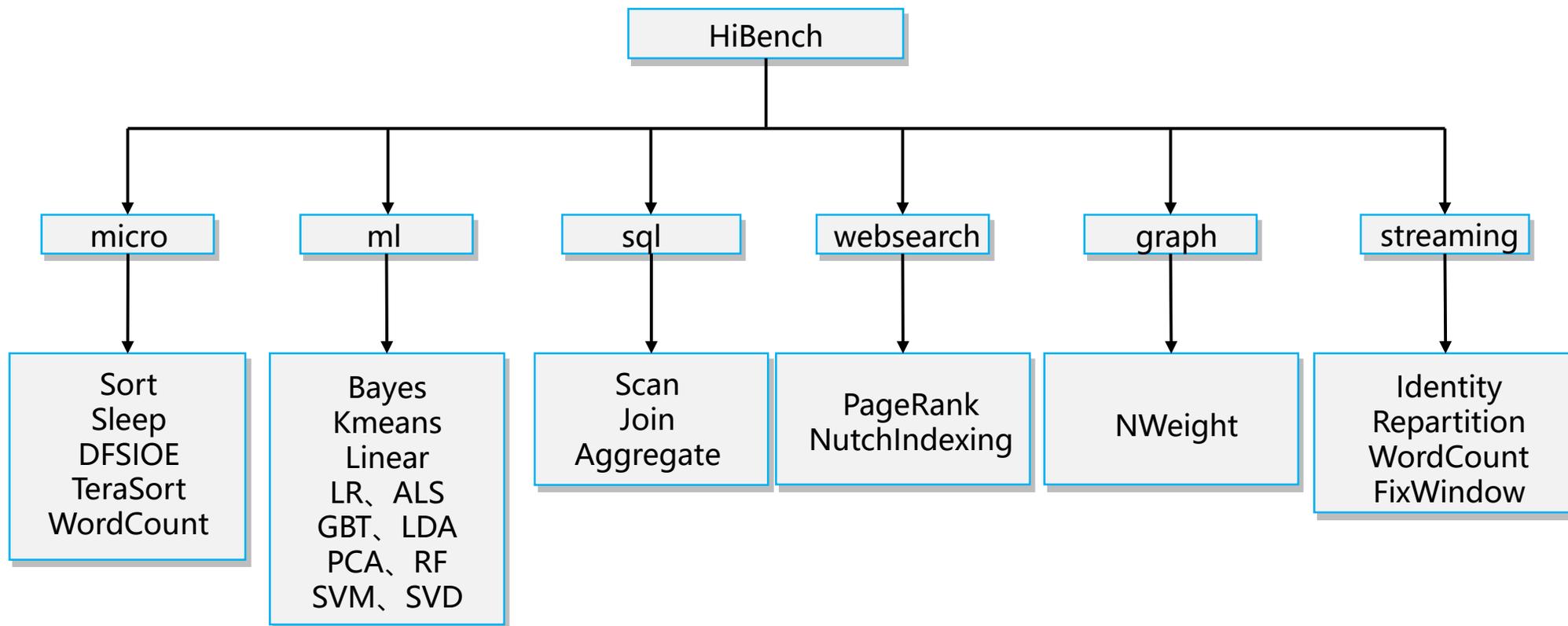
- HiBench简介
 - 开源的大数据基准测试套件。
 - 用于评估大数据框架的速度，吞吐量和系统资源利用率。
 - 支持的框架：hadoopbench、sparkbench、stormbench、flinkbench、gearpumpbench。
 - 支持的开源版本组件：Hadoop、Spark、Storm、Kafka、Flink。
 - 目前使用较多的版本：HiBench-7.0。
 - 共计19个测试方向，可分为6种测试类别。



HiBench - 测试分类

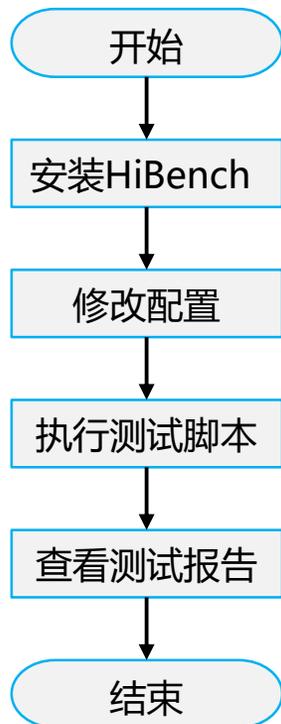
- HiBench测试分类

- 共6种测试类别: micro, ml (机器学习), sql, graph, websearch, streaming。





HiBench - 测试流程



1 安装HiBench

步骤	命令
下载HiBench	Github 地址： https://github.com/intel-hadoop/HiBench
编译HiBench	1. 安装maven 2. 编译指定模块： <code>mvn -Phadoopbench -Psparkbench clean package</code> 3. 直接使用编译好的HiBench

2 创建并配置hadoop.conf

```
# Hadoop home
hibench.hadoop.home /opt/hadoop-3.0.0
# The path of hadoop executable
hibench.hadoop.executable /opt/hadoop-3.0.0/bin/hadoop
# Hadoop configuration directory
hibench.hadoop.configure.dir /opt/hadoop-3.0.0/etc/hadoop
# The root HDFS path to store HiBench data
hibench.hdfs.master hdfs://hadoopmaster:9820
# Hadoop release provider. Supported value: apache, cdh5, hdp
hibench.hadoop.release apache
```

3 执行测试脚本 (以wordcount为例)

脚本名称	作用
prepare.sh	生成测试数据
run.sh	运行脚本

4 查看运行报告

报告名称	作用
hibench.report	存放测试结果



HiBench测试 - 修改配置

- 以Spark WordCount测试项为例，说明HiBench的常规测试流程：

- 步骤一：修改配置文件

- 1. 修改`${HiBench}/conf/hadoop.conf`:

```
# 设置hadoop安装目录（注：因个人环境而异）
hibench.hadoop.home                ${hadoop_home}

# 设置hadoop执行目录
hibench.hadoop.executable          ${hibench.hadoop.home}/bin/hadoop

# 设置hadoop配置目录
hibench.hadoop.configure.dir       ${hibench.hadoop.home}/etc/hadoop

# 设置HDFS root路径，用于存储HiBench数据
hibench.hdfs.master                hdfs://hacluster
```

- 2. 修改`${HiBench}/conf/spark.conf`

```
# 设置spark安装目录（注：因个人环境而异）
hibench.spark.home                 ${spark_home}
```



HiBench测试 - 生成数据

- 步骤二：生成测试数据

- 1. 修改\${HiBench}/conf/hibench.conf:

```
# 设置数据规模，默认规模有tiny、small、large、huge、gigantic、bigdata
# 多种规模对应的数据量含义与测试项强相关，配置文件位于${HiBench}/conf/workloads
hibench.scale.profile          bigdata
```

WordCount对应配置文件为\${HiBench}/conf/workloads/micro/wordcount.conf

```
#datagen
hibench.wordcount.tiny.datasize      32000
hibench.wordcount.small.datasize     320000000
hibench.wordcount.large.datasize     3200000000
hibench.wordcount.huge.datasize      32000000000
hibench.wordcount.gigantic.datasize  320000000000
hibench.wordcount.bigdata.datasize   1600000000000

hibench.workload.datasize            ${hibench.wordcount.${hibench.scale.profile}.datasize}
```

- 2. 生成测试数据

```
# 执行prepare.sh生成数据
bash ${HiBench}/bin/workloads/micro/wordcount/prepare/prepare.sh
```



HiBench测试 - 执行测试

- 步骤三：执行测试

```
# 执行run.sh生成数据
bash ${HiBench}/bin/workloads/micro/wordcount/spark/run.sh
```

- 步骤四：查看测试报告

```
cat ${HiBench}/report/hibench.report
```

Type	Date	Time	Input_data_size	Duration(s)	Throughput(bytes/s)	Throughput/node
ScalaSparkWordcount	2019-09-18	10:08:49	1629155882473	429.012	3797459936	3797459936
ScalaSparkWordcount	2019-09-18	10:20:32	1629155882473	452.070	3603769067	3603769067
ScalaSparkWordcount	2019-09-18	14:36:59	1629155667173	411.862	3955586257	3955586257
ScalaSparkWordcount	2019-09-18	15:22:08	1629156407506	392.421	4151552560	4151552560

主要性能指标：Throughput（吞吐量），数值越高，性能越优。



HiBench测试 - 性能优化思路

- HiBench性能调优与测试大数据组件强相关，以Spark为例：
 - 调整数据分片数、任务并行度（影响数据文件大小和数目，修改参数后，需要重新导入数据）
 - `hibench.default.map.parallelism` # Mapper number in hadoop, partition number in spark
 - `hibench.default.shuffle.parallelism` # Reducer number in hadoop, shuffle partition number in spark
 - 调整Spark Executor数目、CPU核数、内存以及Spark Driver内存大小
 - `hibench.yarn.executor.num` # Executor number
 - `hibench.yarn.executor.cores` # Executor cores
 - `spark.executor.memory` # Executor memory
 - `spark.driver.memory` # Driver memory



思考题

1. HiBench支持的框架有哪些? ()
 - A. flinkbench
 - B. hadoopbench
 - C. stormbench
 - D. sparkbench



本节小结

- 本节主要介绍了大数据的基本概念，并以HiBench为例介绍了在大数据场景下进行性能测试的方法和主要性能指标，以及HiBench基本的性能调优思路。



目录

1. 数据库性能测试
2. 大数据性能测试
3. HPC性能测试
4. 性能调优分析工具
5. 性能调优案例



本节概述和学习目标

- 本节主要以WRF为例，介绍了在HPC场景下进行性能测试的方法。
- 学完本节内容后，您将能够：
 - 理解HPC的基本概念
 - 掌握WRF在鲲鹏平台上的性能测试方法和主要性能指标
 - 描述WRF基本的性能优化思路

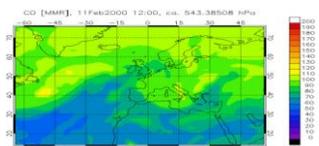


高性能计算简介

- 什么是HPC?

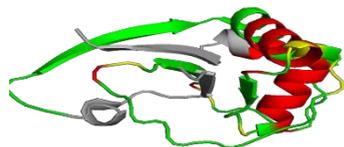
- HPC (High Performance Computing) 高性能计算, 是通过**高速网络**将大量服务器进行互联形成计算机**集群**, 与高性能**存储**一起, 求解科研、工业界最复杂的**科学计算**问题 (科学研究领域三大范式: 理论科学, 实验科学, 计算科学)。

- 典型应用领域:



环境科学

天气/气象预测
污染/臭氧空洞



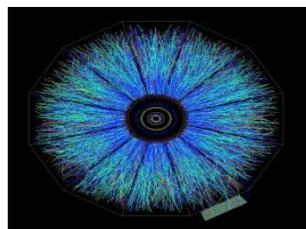
生命科学

医学/生物学
基因测序/蛋白质动力学



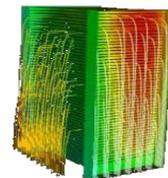
材料学/化学

自旋电子学/纳米科学
分子动力学



天体物理

天体测量/天体力学
宇宙起源



能源

等离子体物理学
油气勘探/燃料电池



制造

CAE仿真/汽车/芯片
碰撞/流体/电磁



HPC典型应用 - WRF

- WRF (Weather Research and Forecasting Model) 是次世代的中尺度天气预报模式，包括多重区域、从几公里到数千公里的灵活分辨率、多重嵌套网格，以及与之协调的三维变分同化系统3DVAR等。WRF数值模式采用高度模块化、并行化和分层设计技术，集成了迄今为止中尺度方面的研究成果。WRF模式不仅可以用于真实天气的个案模拟，也可以用其包含的模块组作为基本物理过程探讨的理论根据，包括如下内容：
 - 气象研究
 - 实时NWP
 - 理想模拟
 - 数据同化
 - 地球系统模型耦合
 - 示范训练和教育支持



HPC典型应用 - WRF模拟系统

- WRF模拟系统主要包含WPS和WRF两部分模块：
 - WPS模块：全称为WRF Pre-processing System，即WRF预处理系统，用来为WRF模型准备输入数据；如果只是做理想实验(idealized modeling)，就不需要用WPS处理真实数据。
 - WRF模块：就是数值求解的模块，它有两个版本：ARW(Advanced Research WRF) 和 NMM(Nonhydrostatic Mesoscale Model)。



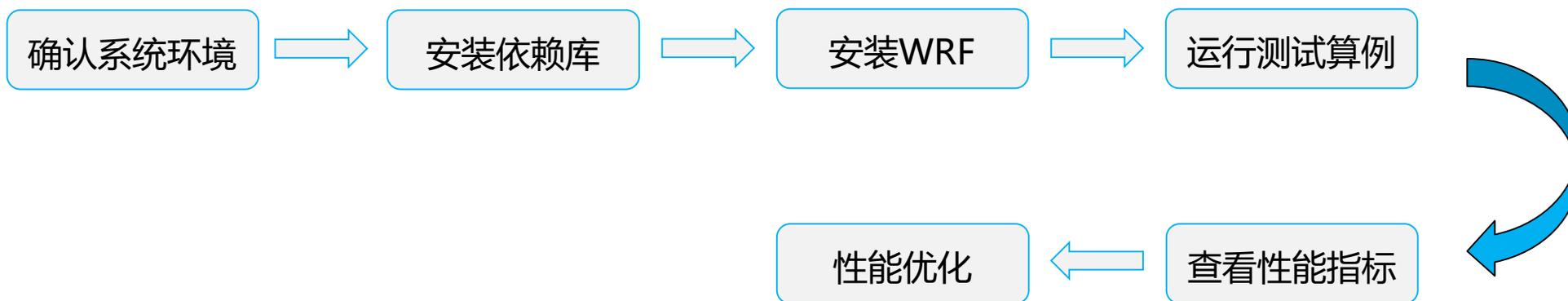
WRF安装环境推荐配置

类别	子类	要求	软件描述
硬件	TaiShan X6000 XA320 V2 服务器	CPU: 2*Kunpeng 920-4826	-
		内存: 16*16GB 2666MHz RAM	-
		硬盘: 1*300GB SAS HD	-
OS	CentOS	CentOS-7.6 aarch64	-
软件	WRF	3.8.1	开源气象模拟软件
	GCC	9.1	开源编译器
	OpenMPI	4.0.1	开源并行库
	HDF5	1.10.1	层级数据格式
	NETCDF	4.4.1.1	网络通用数据格式
	NETCDF-F	4.4.1	NETCDF fortran库
	PNETCDF	1.8.0	并行NETCDF支持MPI I/O



性能测试 - WRF测试流程

- 如下是基于华为鲲鹏平台以气象行业应用WRF为例，进行单次安装测试的流程图：





性能测试 - 确认系统环境

- CentOS 7.6常用命令：

命令	作用	命令	作用
env	查看环境变量	pwd	查看当前位置路径
ls	查看文件	mv	重命名或移动
cd	进入路径	cp	复制
rm	删除	yum/apt-get	软件包管理器

- 确认环境网络配置信息：
 - 环境是否可连接外网、可用网络源及代理设置、关闭防火墙、节点配置信息、高速网络类型等。
 - 如果使用InfiniBand (IB)网络确认是否安装加速库并获得加速库安装路径信息。



性能测试 - 安装依赖库

- 步骤一：安装编译器或确认已安装的编译器版本信息，以GNU编译器为例：
 - Gcc、g++、gfortran
- 步骤二：按顺序安装MPI及下列依赖库：
 - OpenMPI \implies HDF5 \implies PNETCDF \implies NETCDF-C \implies NETCDF-F
- 步骤三：在安装过程中推荐使用环境变量来管理依赖库地址，例如：

```
# export NETCDF=/path/to/netcdf
# export CPPFLAGS="-I$NETCDF/include"
# export LDFLAGS="-L$NETCDF/lib -lnetcdf -lz"
```

- 步骤四：大部分依赖库可以通过以下命令完成安装

```
# ./configure && make && make install
```



性能测试 - 安装WRF

- 步骤一：解压安装包，进入源码目录下
- 步骤二：修改arch/configure_new.defaults，添加鲲鹏架构编译参数：

```
#ARCH      Linux      aarch64, gnu OpenMPI #serial smpar dmpar dm+sm
DESCRIPTION =          GNU ($SFC/$SCC)
.....
DM_CC      =          mpicc -cc=$(SCC) -L /home/armmath/lib -lmathlib
.....
CFLAGS_LOCAL =        -w -O3 -c -mcpu=tsv110
.....
CC_TOOLS   =          $(SCC)
```

- 步骤三：执行./configure生成配置文件，并按提示输入如下两个选项：4（回车）和1（回车）

```
-----
Please select from among the following Linux aarch64 options:

  1. (serial)  2. (smpar)  3. (dmpar)  4. (dm+sm)  GNU (gfortran/gcc)

Enter selection [1-4] : 4
-----
Compile for nesting? (1=basic, 2=preset moves, 3=vortex following) [default 1]: 1
```

- 步骤四：执行命令进行编译安装

```
# ./compile -j 16 wrf 2>&1 | tee -a compile.log
```



性能测试 - 运行测试算例

- 步骤一：准备算例文件
 1. 下载官网提供的公开算例Conus 2.5km的三个必要文件：
 - 边界条件数据文件：wrfbdy_d01.gz
 - 续算文件：rst_6hr*
 - 算例配置文件：namelist.input

2. 执行命令生成算例的数据文件

```
# cat rst_6hr* | gunzip -c > wrfrst_d01_2005-06-04_06_00_00
# gunzip wrfbdy_d01.gz
```

3. 在手动下载数据文件的情况下，需要在配置文件namelist.input中的“&dynamics”部分增加如下一行：

```
&dynamics
w_damping =          1,
diff_opt =          1,
.....
non_hydrostatic =    .true.,
use_baseparam_fr_nml = .t.,
/
```



性能测试 - 查看性能指标

- 步骤二：将编译生成的执行文件拷贝到算例目录下

```
# cp /path/to/WRF-3.8.1/run/* /path/to/conus_2.5km/
```

- 步骤三：执行命令运行算例

```
# echo 3 > /proc/sys/vm/drop_caches          #释放内存空间，以保证WRF的运行性能  
# time mpirun --allow-run-as-root -np 32 ./wrf.exe
```

- 步骤四：查看“rsl.error.0000”日志中的“elapsed seconds”数值，并对比基线数据，数值越低性能越优

```
Timing for main: time 2005-06-04_06:00:45 on domain 1: 4.16054 elapsed seconds  
Timing for main: time 2005-06-04_06:01:00 on domain 1: 4.16418 elapsed seconds  
Timing for main: time 2005-06-04_06:01:15 on domain 1: 4.15825 elapsed seconds  
Timing for main: time 2005-06-04_06:01:30 on domain 1: 4.15202 elapsed seconds  
Timing for main: time 2005-06-04_06:01:45 on domain 1: 4.16020 elapsed seconds  
Timing for main: time 2005-06-04_06:02:00 on domain 1: 4.15511 elapsed seconds  
Timing for main: time 2005-06-04_06:02:15 on domain 1: 4.14857 elapsed seconds  
Timing for main: time 2005-06-04_06:02:30 on domain 1: 4.15593 elapsed seconds
```



性能测试 - 性能优化思路

- 运行算例时使用MPI参数设置OpenMP线程或通过绑核提高WRF运算性能。
- 更换AHC编译器和optimized-routines-master数学库重新编译WRF，运行算例，对比性能数据。
- 使用应用特征分析工具对WRF进行分析，根据分析结果对造成阻塞的点进行优化来提升性能。



思考题

1. 下列哪些选项可能会影响WRF性能 ()
 - A. 网络带宽
 - B. 并行线程数
 - C. 内存刷新频率
 - D. 存储读写速度



本节小结

- 本节主要介绍了HPC的基本概念，并以WRF为例介绍了在HPC场景下进行性能测试的方法和主要性能指标，以及WRF基本的性能调优思路。



目录

1. 数据库性能测试
2. 大数据性能测试
3. HPC性能测试
4. 性能调优分析工具
 - 华为鲲鹏性能优化工具介绍
 - C/C++程序性能分析与优化
 - Java Mixed-Mode程序性能分析与优化
5. 性能调优案例



本节概述和学习目标

- 本节主要介绍了华为鲲鹏性能优化工具的定位、功能特性、应用场景、逻辑架构、访问方式和业务流程，以及如何使用该工具对C/C++类程序和Java Mixed-Mode类程序进行性能分析与优化的方法。
- 学完本节内容后，您将能够：
 - 了解华为鲲鹏性能优化工具的定位、应用场景、支持的功能特性和访问方式等
 - 掌握C/C++类程序性能分析的方法
 - 掌握Java Mixed-Mode程序性能分析的方法



性能调优概述

性能调优

- 性能调优就是对计算机硬件、操作系统和应用程序有相当深入的了解，调节三者之间的关系，实现整个系统（包括硬件、操作系统、应用程序）的性能最大化，并能不断地满足现有的业务需求。



为什么要调优?

一是为了获得更好的系统性能。

二是通过性能调优来满足不断增加的业务需求。



硬件调优

CPU、内存、磁盘、网卡.....



操作系统调优

进程、文件系统、内核.....



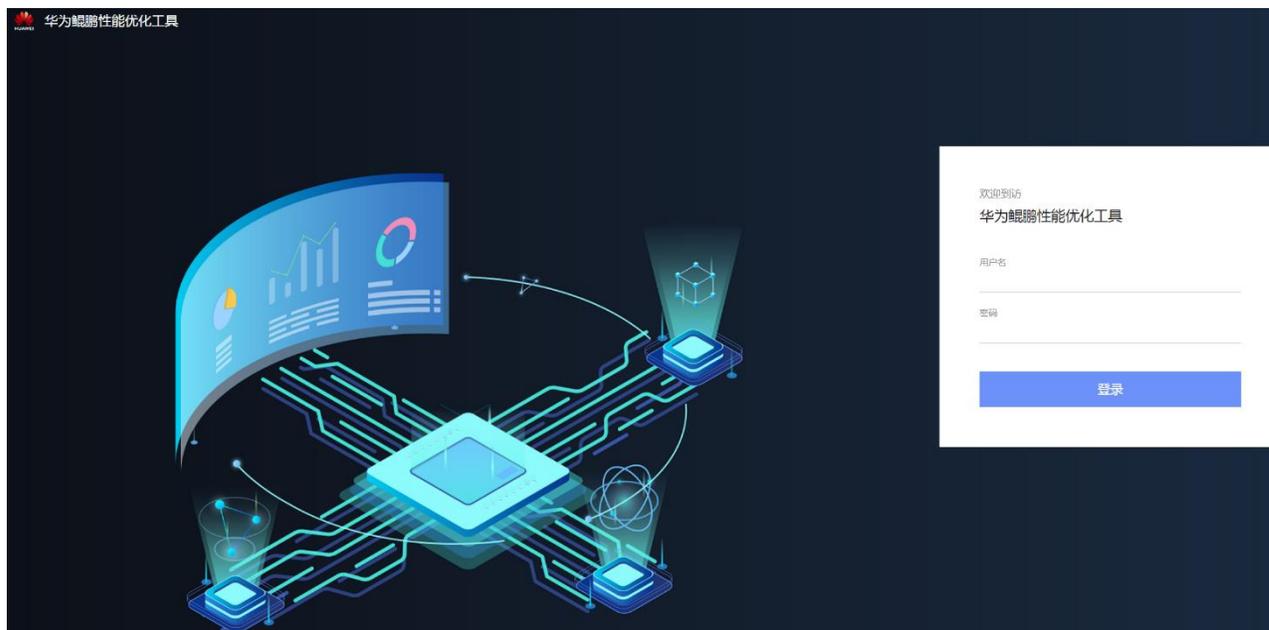
应用程序调优

Nginx、MySQL.....



华为鲲鹏性能优化工具是什么

- 为解决客户软件运行遇到性能问题时凭人工经验定位困难、调优能力弱的痛点，华为推出了Kunpeng Tuning Kit鲲鹏性能优化工具。
- 华为鲲鹏性能优化工具主要面向华为FAE、开放实验室能力建设工程师或客户工程师，针对应用程序部署在TaiShan服务器的场景下，通过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，定位到瓶颈点及热点函数。





华为鲲鹏性能优化工具功能特性



支持采集整个系统或指定进程的CPU Cycles性能事件，能够快速定位热点函数。



支持热点函数按照CPU核/线程/模块进行分组，支持查看热点函数调用栈。



支持通过火焰图查看热点函数及其调用栈。



支持代码映射功能，即查看函数内的热点指令及该指令对应的高级语言文件及行号。



支持显示汇编代码的控制流图。



支持分析Java代码的热点函数及热点指令。



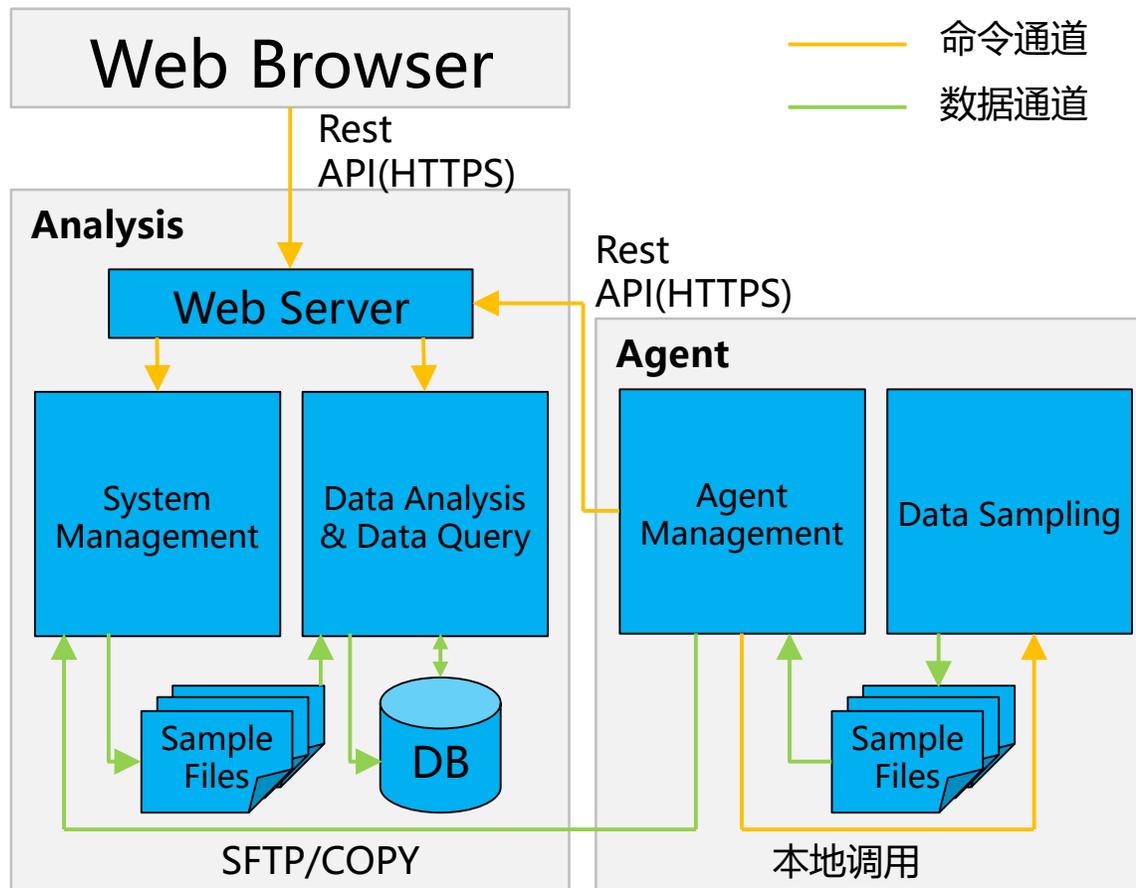
华为鲲鹏性能优化工具应用场景

- 当客户的应用软件运行在TaiShan服务器上时，如果遇到性能或体验问题，可通过华为鲲鹏性能优化工具来快速分析、定位及调优。



华为鲲鹏性能优化工具逻辑架构

- 华为鲲鹏性能优化工具从逻辑上分成 Analysis 和 Agent 两大部分：
 - Analysis：主要作用是实现性能数据分析及分析结果呈现。
 - Agent：主要作用是实现性能数据采集。





华为鲲鹏性能优化工具子模块

Analysis子模块	功能说明
Web Browser	Web浏览器，用于操作交互和数据呈现。
Web Server	Web服务器。
System Management	提供用户管理、数据库存储空间管理、日志管理、采集分析任务管理等功能。
Data Analysis & Query	<ul style="list-style-type: none">● 对采集数据文件预处理，导入数据库中。● 根据分析请求，将原始数据汇总，结合以往项目调优经验值，完成对性能数据的分析，给出优化建议。● 提供性能分析结果查询接口。

Agent子模块	功能说明
Agent Management	提供获取采集任务、采集文件存储空间管理、日志管理等功能。
Data Sampling	<ul style="list-style-type: none">● 完成对系统的性能数据采集，并保存到文件。● 将文件通过SFTP/COPY上传到Analysis。



华为鲲鹏性能优化工具部署方式

- 当前版本**只支持单机部署**，即将华为鲲鹏性能优化工具所有组件部署在一台服务器上，完成对该台服务器软件的性能数据采集和分析。



将Analysis和Agent部署在一台服务器上

- 部署环境要求如下表所示。

类别	子类	要求
硬件	服务器	TaiShan 200服务器，采用华为鲲鹏920处理器
操作系统	CentOS	CentOS 7.6，内核版本要求4.14.0以上
	openEuler	openEuler开源社区版本



华为鲲鹏性能优化工具访问方式

- 华为鲲鹏性能优化工具部署在TaiShan服务器上，该服务器上同时运行客户的应用软件。
- 华为鲲鹏性能优化工具提供Web界面访问方式，用户只需要在浏览器地址栏中输入：
https://部署服务器的IP:端口号即可。
- 访问Web界面时，对本地浏览器的要求如下表所示。

浏览器类型	要求
Google Chrome	Chrome 72.0及以上版本
Internet Explorer	Internet Explorer 10.0及以上版本
Firefox	Firefox 62.0及以上版本



华为鲲鹏性能优化工具业务流程



输入

- 创建性能分析任务
- 配置任务属性参数（分析类型、应用路径、CPU采样周期等）
- 运行待分析软件和分析任务



分析处理

- 采集处理器性能指标数据
- 采集函数（C/C++/Java）性能指标数据
- 将采集数据文件按不同指标维度数据库化保存
- 统计分析，对比经验指标，定位出性能瓶颈



输出

- Top热点函数
- 热点代码块（源码&汇编展示）
- 火焰图展示函数间调用关系



目录

1. 数据库性能测试
2. 大数据性能测试
3. HPC性能测试
4. 性能调优分析工具
 - 华为鲲鹏性能优化工具介绍
 - C/C++程序性能分析与优化
 - Java Mixed-Mode程序性能分析与优化
5. 性能调优案例



C/C++ 程序性能分析和优化

01

创建工程

创建工程，用于统一管理该工程下的所有性能分析任务。

03

启动性能分析

如果创建分析任务时未自动启动，则可手动启动分析任务。

02

创建C/C++分析任务

根据应用程序的类型，选择创建C/C++类性能分析任务。

04

查看分析结果

查看采集日志、分析结果统计总览、热点函数、函数代码映射、函数火焰图。



创建工程和分析任务

- 步骤一：登录华为鲲鹏性能优化工具Web界面。
 - 在浏览器地址栏输入：`https://部署服务器的IP:8086`，（例如：`https://10.116.239.242:8086`），默认用户名为 **malluma**，默认密码为 **Huawei12#\$**。
- 步骤二：创建工程。
- 步骤三：在已创建的工程下，单击“函数分析任务”。
- 步骤四：从下拉列表中选择“新建分析任务 > C/C++程序”，配置任务相关的参数，参数说明见下一页。
- 步骤五：创建分析任务后可以选择“立即执行”，或者根据应用程序的启动情况，后续手动启动分析任务。





创建工程和分析任务 - 参数说明

参数名称	参数说明
任务名称	新建分析任务的名称。
分析类型	<ul style="list-style-type: none">● Launch Application: 采集启动的时候, 同时启动Application, 采集时长受Application的执行时间来控制, 适用于Application运行时间较短的场景。● Profile System: 采集整个服务器系统, 无需关注系统中有哪些类型的应用在运行, 采集时长需要配置参数控制, 适用于多业务混合运行和有子进程的场景。● Attach to Process: 该采集对象针对正在运行的应用程序, 通过关联到运行该应用程序的PID的方式, 来实时跟踪和采集该应用程序的性能数据。采集时长需要配置参数控制, 适用于某些应用需要长时间持续运行和采集的场景。
应用	输入待分析的目标应用在服务器上的绝对路径, 需要加上文件名称。
应用参数	输入应用程序执行时需要的参数。
工作目录	可以选择与应用路径一致, 也可以自定义, 自定义时需要输入Application的运行绝对路径。
PID	当分析类型选择“Attach to Process”时, 需要设置应用程序进程的PID, 以便实时跟踪和采集该应用程序的性能数据。
采集时长	设置采集的时长。分析类型选择“Profile System”或“Attach to Process”时需配置此参数。
CPU采集周期	设置CPU采集周期, 可以选择自定义或高精度。高精度采样针对CPU 1ms采样精度数据不够精确的场景, 能够准确跟踪应用程序在高频CPU上运行所占用所有微指令CPU running time (CPU运行时间), 如Frontend Bound、Backend Bound各指标等。
待采集CPU核	针对某些CPU核进行性能分析的场景, 如需要观察某应用程序在某个CPU核上的性能数据, 需要在此输入对应的CPU核, 取值范围是0 ~ (服务器总CPU核数 - 1)。例如: 如CPU 16核的服务器, 可以输入“0,1,2,10”, 表示针对CPU0, CPU1, CPU2, CPU10进行性能数据采集和分析。
二进制/符号文件路径	输入二进制/符号文件在服务器上的绝对路径。当开发者需要观察源代码和汇编指令映射后的性能数据时, 并且对应的应用程序无符号表信息, 该参数用来导入对应应用程序的符号表。
C/C++源文件路径	输入源文件在服务器上的绝对路径。该应用的源码请务必符合通用编程规范, 否则分析结果中热点函数的源码可能无法正常显示。当开发者需要观察源代码和汇编指令映射后的性能数据时, 该参数用来导入对应应用程序的源代码。



启动分析任务

- 如果创建分析任务时，已勾选“立即执行”，则工具会自动执行该分析任务。否则，请按照本节操作手动启动分析任务。
- 步骤一：单击已创建工程下的“函数分析任务”。
- 步骤二：选择已创建的C/C++性能分析任务。单击任务操作列的“启动”，启动性能分析任务。



< | NEW_01

新建分析任务 搜索

任务名称	任务类型	任务状态	执行时间(秒)	分析类型	创建时间	操作
new_analysis_01	C/C++ Program	Created	--	Launch Application	2019-09-03 08:53:42	启动 修改 删除

10 总条数: 1 < 1 >



查看采集日志

- 性能分析任务启动后，可以通过系统打印出来的采集日志信息，及时获取采集成功的状态和解析的日志信息，以及采集失败或解析失败的报错信息。
 - 步骤一：单击已创建工程下的“函数分析任务”。
 - 步骤二：单击已创建的C/C++性能分析任务名称，进入任务详情界面。
 - 步骤三：单击“采集日志”页签，查看采集过程的状态信息、日志信息、数据分析信息。



test_001

配置信息 | 采集日志 | 总览 | 函数 | 火焰图

采集过程

```
[Fri 06 Sep 2019 10:38:17] [CollectionEngine.py collect_preparation] -[line:1055] [INFO] sudo perf record -o /opt/malluma/data/sample/testbox/test_001/data/perf.data -F 1000 -N -B -T -g -e [cycles,r008,r01b,r10c,r009,r010,r012],[cycles,r100,r102,r107,r108,r076,r078],[cycles,r001,r002,r014,r179,r177],[cycles,r121,r122,r123,r124,r125,r126],[cycles,r040,r042,r050,r052,r060,r061],[cycles,r003,r004,r005,r016,r017],[cycles,r070,r071,r073,r074,r075,r077],[cycles,r112,r113,r11z,r111,r120],[cycles,r06c,r06d,r06e,r07c,r07d,r07e],[cycles,r150,r151,r152,r16a,r079,r07a] /home/chenox/temp/Test_Malluma

[Fri 06 Sep 2019 10:38:17] [CollectionEngine.py change_perfevent_maxsample_rate] -[line:1002] [INFO] sudo sh -c 'echo 100000 > /proc/sys/kernel/perf_event_max_sample_rate'

[Fri 06 Sep 2019 10:38:17] [CollectionEngine.py change_perfevent_maxsample_rate] -[line:1007] [INFO] perf event max sample rate changed successful.

[Fri 06 Sep 2019 10:38:32] [CollectionEngine.py collection_trigger] -[line:1351] [INFO] General Exploration Sampling finished.

[Fri 06 Sep 2019 10:38:32] [CollectionEngine.py get_stop] -[line:694] [INFO] Elapsed Time: begin=1264193.56, end=1264209.03, end-begin=15.47

[Fri 06 Sep 2019 10:38:32] [CollectionEngine.py collect_start] -[line:1069] [INFO] Collection finished
```

数据分析

```
[Fri 06 Sep 2019 10:38:32] [CollectionEngine.py resolve_origin_data] -[line:1080] [INFO] perf script -F comm,pid,time,period,event,ip,sym,dso,symoff -i /opt/malluma/data/sample/testbox/test_001/data/perf.data --show-kernel-path

[Fri 06 Sep 2019 10:38:37] [CollectionEngine.py resolve_perf_data] -[line:878] [INFO] Perf data resolve finished.

[Fri 06 Sep 2019 10:38:37] [CollectionEngine.py resolve_perf_data] -[line:883] [INFO] Malluma collection finished.

[Fri 06 Sep 2019 10:38:38] [MultiProcessCB.py multiprocessing_callback] -[line:189] [INFO] process 11167 end insert OriginalData

[Fri 06 Sep 2019 10:38:38] [MultiProcessCB.py multiprocessing_callback] -[line:189] [INFO] process 11163 end insert OriginalData

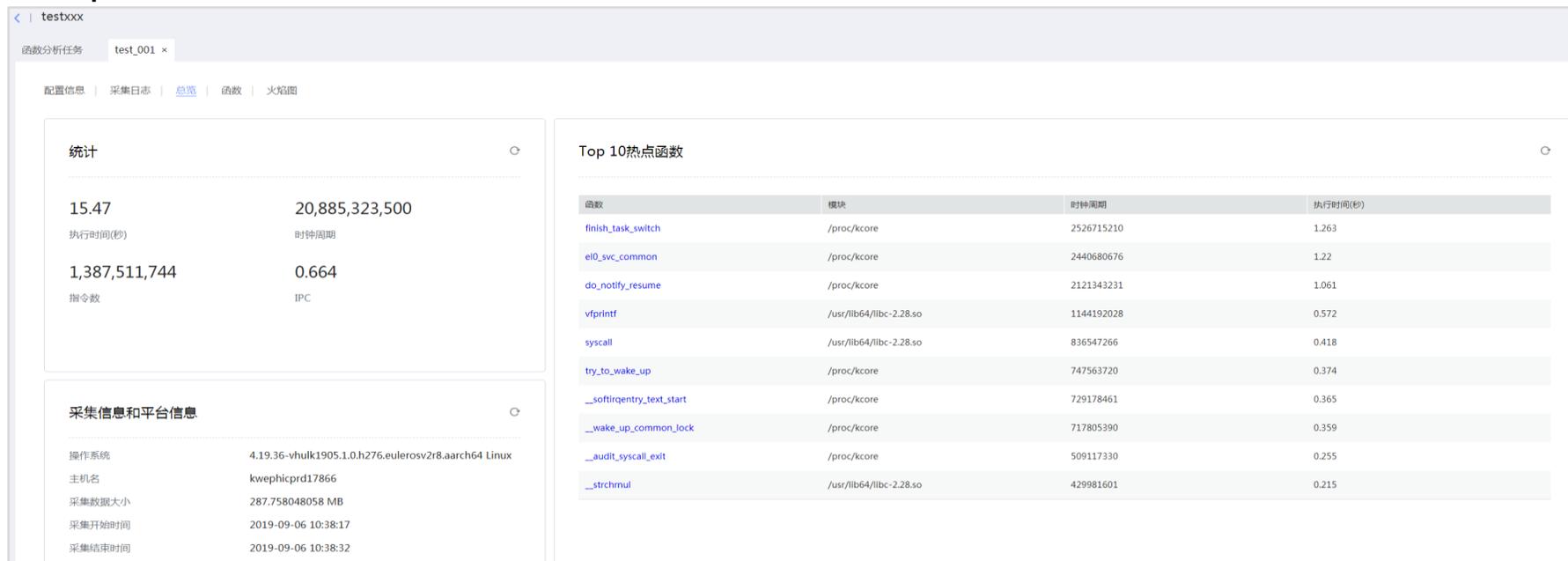
[Fri 06 Sep 2019 10:38:39] [MultiProcessCB.py multiprocessing_callback] -[line:189] [INFO] process 11164 end insert OriginalData

[Fri 06 Sep 2019 10:38:39] [MultiProcessCB.py multiprocessing_callback] -[line:189] [INFO] process 11162 end insert OriginalData
```



查看分析结果总览

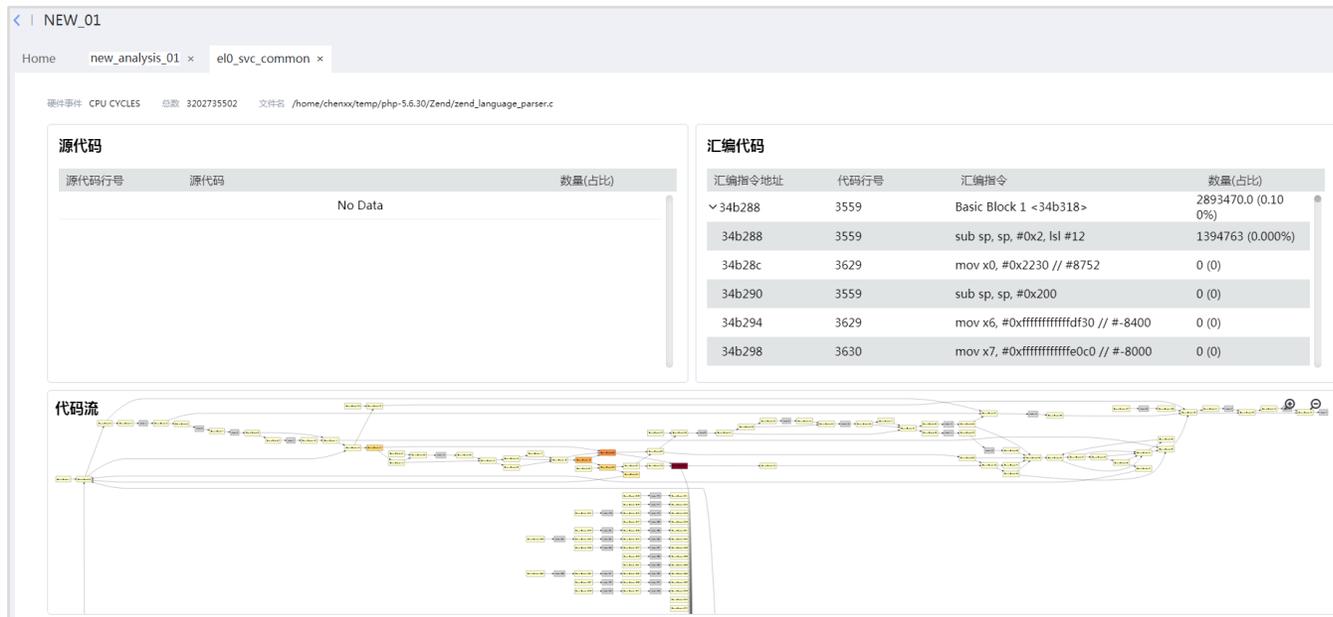
- 性能分析任务完成后，可以查看该任务的分析结果。
 - 步骤一：单击已创建工程下的“函数分析任务”。
 - 步骤二：单击已创建的C/C++性能分析任务名称，进入任务详情界面。
 - 步骤三：单击“总览”页签，查看采集后的指标统计信息、采集平台信息、采集信息、Top 10函数信息。





查看热点函数分析结果

- 通过查看热点函数的代码映射，可以分析热点函数内部的热点指令，热点指令即指函数内Cycles事件占比最高的Top指令。同时还可以查看热点指令对应的高级语言文件及行号，以及对汇编代码进行控制流分析。
 - 步骤一：单击已创建工程下的“函数分析任务”。
 - 步骤二：单击已创建的C/C++性能分析任务名称，进入任务详情界面。
 - 步骤三：在Top10函数区域单击某个热点函数，查看该函数的代码映射情况。
 - 查看硬件事件类型、该事件的求和值、函数所在的源文件。
 - 查看函数源代码、函数汇编代码和函数汇编代码basic block的代码流图。





查看所有函数分析结果

- 通过分析不同函数对应top-down模型的各指标值，按选定指标排序查看其TopN热点函数，可知针对某些硬件指标哪些函数是主要瓶颈。
 - 步骤一：单击已创建工程下的“函数分析任务”。
 - 步骤二：单击已创建的C/C++性能分析任务名称，进入任务详情界面。
 - 步骤三：单击“函数”页签，查看所有函数的各指标统计信息。



- Module/Function/Callstack
- Function/Callstack View
- Module/Function/Callstack**
- Thread/Function/Callstack
- Core/Function/Callstack

Function/Callstack	执行时间(秒)	时钟周期	指令数	IPC	模块
▼Total	10.44266175	20885323500	1387511744	0.664347738736247	
> finish_task_switch	1.263357605	2526715210	188409902	0.7456713018322314	/proc/kcore
> elf_svc_common	1.220340338	2440680676	148068780	0.6066700222442373	/proc/kcore
> do_notify_resume	1.0606716155	2121343231	168427612	0.7939668109276372	/proc/kcore
> vforintf	0.572096014	1144192028	85844988	0.7502673143952372	/usr/lib64/libc-2.28.so
> syscall	0.418273633	836547266	53135022	0.6351705893926143	/usr/lib64/libc-2.28.so
> try_to_wake_up	0.37378186	747563720	30708097	0.4107756459877427	/proc/kcore
> _softirqentry_text_start	0.3645892305	729178461	79926246	1.0961136439821417	/proc/kcore
> _wake_up_common_lock	0.358902695	717805390	19127073	0.2664659985347839	/proc/kcore
> __audit_syscall_exit	0.254558665	509117330	30124221	0.5916950617257519	/proc/kcore

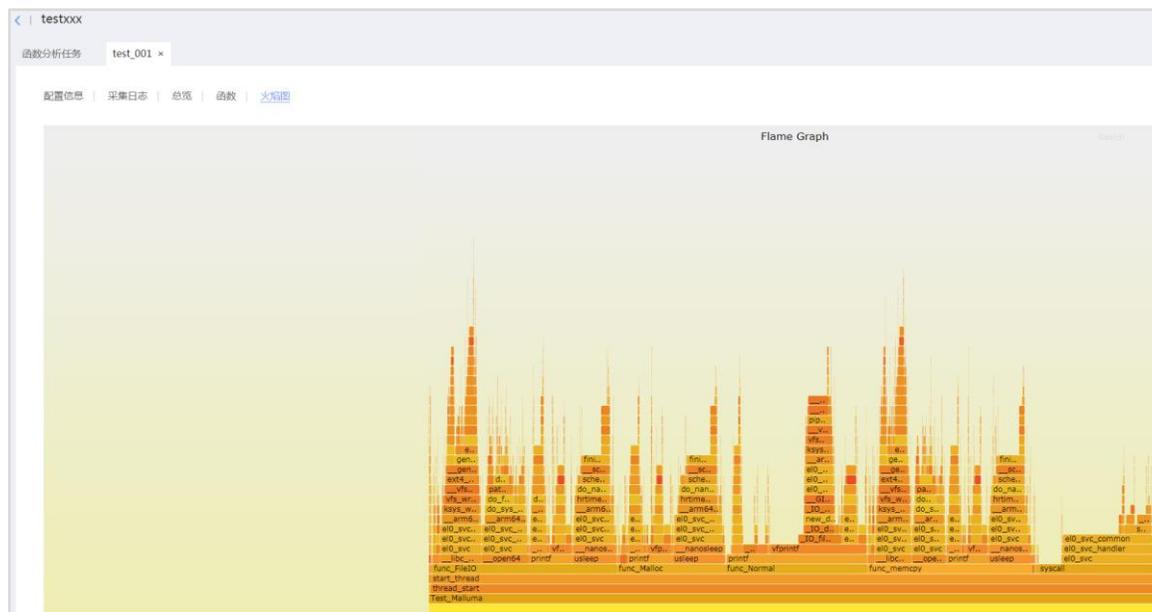


查看函数火焰图

- 通过查看采样时间热点函数的火焰图，帮助用户直观且迅速找到热点代码路径。

- 步骤一：单击已创建工程下的“函数分析任务”。
- 步骤二：单击已创建的C/C++性能分析任务名称，进入任务详情界面。
- 步骤三：单击“火焰图”页签，查看函数的火焰图。
- 步骤四：火焰图操作技巧。

- 单击某个函数，可以放大显示该函数的调用栈。
- 单击“Reset Zoom”即可恢复原火焰图。
- 在火焰图右上角“Search”区域输入函数名称，可以快速定位到该函数。
- 单击“Reset Search”即可恢复原火焰图。





目录

1. 数据库性能测试
2. 大数据性能测试
3. HPC性能测试
4. 性能调优分析工具
 - 华为鲲鹏性能优化工具介绍
 - C/C++程序性能分析与优化
 - Java Mixed-Mode程序性能分析与优化
5. 性能调优案例



Java Mixed-Mode程序性能分析和优化

01

创建工程

创建工程，用于统一管理该工程下的所有性能分析任务。

03

启动性能分析

手动启动分析任务。

02

创建Java Mixed-Mode分析任务

根据应用程序的类型，选择创建Java类程序性能分析任务。

查看分析结果

查看采集日志、分析结果统计总览、热点函数、函数代码映射、函数火焰图。

04



创建工程和分析任务

- 步骤一：登录华为鲲鹏性能优化工具Web界面。
 - 在浏览器地址栏输入：`https://部署服务器的IP:8086`，（例如：`https://10.116.239.242:8086`）。默认用户名为 `malluma`，默认密码为 `Huawei12#$`。
- 步骤二：创建工程。
- 步骤三：在已创建的工程下，单击“函数分析任务”。
- 步骤四：从下拉列表中选择“新建分析任务Java Mixed-Mode程序”，配置任务相关的参数。参数说明见下一页。
- 步骤五：创建分析任务后可以选择“立即执行”，或者根据应用程序的启动情况，后续手动启动分析任务。



新建Java Mixed-Mode分析任务

* 任务名称

分析类型
Launch Application即采集启动的时候同时启动Application，采集时长受Application的执行时间来控制，适用于Application运行时间较短的场景。

* 应用

* 应用参数

* 工作目录

* CPU采集周期(毫秒)

Java源文件路径

立即执行



创建工程和分析任务 - 参数说明

参数名称	参数说明
任务名称	新建分析任务的名称。
分析类型	对于Java Mixed-Mode类型的程序，支持以下两种分析类型： <ul style="list-style-type: none">● Launch Application：采集启动的时候同时启动Application，采集时长受Application的执行时间来控制，适用于Application运行时间较短的场景。● Attach to Process：该采集对象针对正在运行的应用程序，通过关联到运行该应用程序的PID的方式，来实时跟踪和采集该应用程序的性能数据。采集时长需要配置参数控制，适用于某些应用需要长时间持续运行和采集的场景。
应用	输入待分析的应用程序在服务器上的绝对路径。
应用参数	输入应用程序运行时自身的参数。
工作目录	可以选择与应用路径一致，也可以自定义，自定义时需要输入Application的运行绝对路径。
PID/TID	当分析类型选择“Attach to Process”时，需要设置应用程序进程的PID，以便实时跟踪和采集该应用程序的性能数据。
采集时长	设置采集的时长。
CPU采集周期	设置CPU采集周期，可以选择自定义或高精度。高精度采样针对CPU 1ms采样精度数据不够精确的场景，能够准确跟踪应用程序在高频CPU上运行所占用所有微指令CPU running time（CPU运行时间），如Frontend Bound、Backend Bound各指标等。
Java源文件路径	输入Java源文件所在路径。该应用的源码请务必符合通用编程规范，否则分析结果中热点函数的源码可能无法正常显示。当开发者需要观察源代码映射后的性能数据时，该参数用来导入对应应用程序的源代码。



启动分析任务

- 如果创建分析任务时，已勾选“立即执行”，则工具会自动执行该分析任务。否则，请按照本节操作手动启动分析任务。
 - 步骤一：单击已创建工程下的“函数分析任务”。
 - 步骤二：选择已创建的Java Mixed-Mode性能分析任务。单击任务操作列的“启动”，启动性能分析任务。



< | NEW_01

新建分析任务 搜索

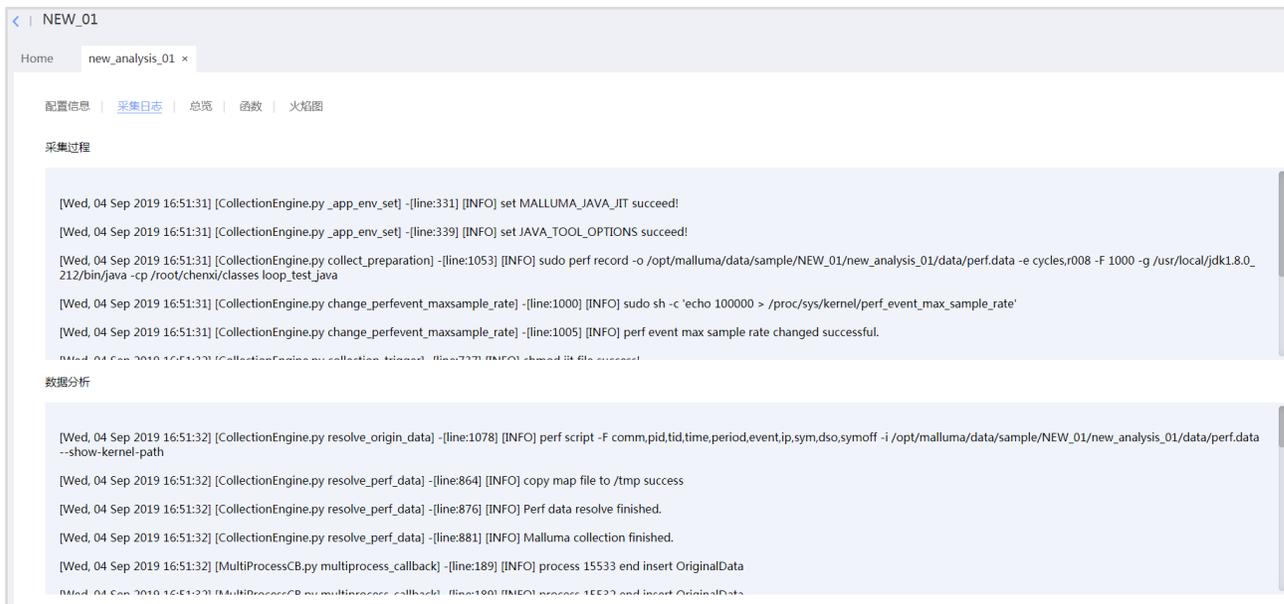
任务名称	任务类型	任务状态	执行时间(秒)	分析类型	创建时间	操作
new_analysis_01	java-mixed-mode	Created	--	Launch Application	2019-09-02 14:10:39	启动 修改 删除

10 总条数: 1 < 1 >



查看采集日志

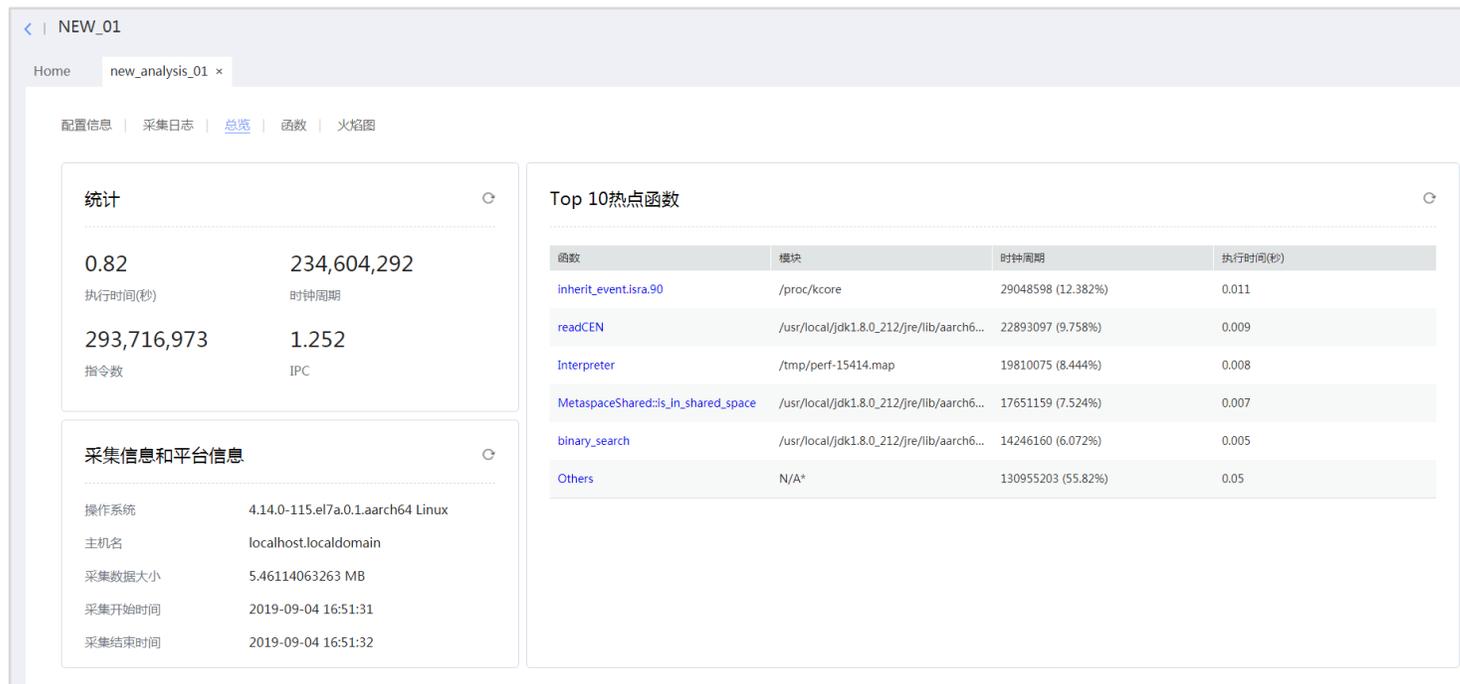
- 性能分析任务启动后，可以通过系统打印出来的采集日志信息，及时获取采集成功的状态和解析的日志信息，以及采集失败或解析失败的报错信息。
 - 步骤一：单击已创建工程下的“函数分析任务”。
 - 步骤二：单击已创建的Java Mixed-Mode性能分析任务名称，进入任务详情界面。
 - 步骤三：单击“采集日志”页签，查看采集过程的状态信息、日志信息、数据分析信息。





查看分析结果总览

- 性能分析任务完成后，可以查看该任务的分析结果。
 - 步骤一：单击已创建工程下的“函数分析任务”。
 - 步骤二：单击已创建的Java Mixed-Mode性能分析任务名称，进入任务详情界面。
 - 步骤三：单击“总览”页签，查看采集后的指标统计信息、采集平台信息、采集信息、Top 10函数信息。





查看热点函数分析结果

- 通过查看热点函数的代码映射，可以分析热点函数内部的热点指令，热点指令即指函数内Cycles事件占比最高的Top指令。同时还可以查看热点指令对应的高级语言文件及行号，以及对汇编代码进行控制流分析。

- 步骤一：单击已创建工程下的“函数分析任务”。
- 步骤二：单击已创建的Java Mixed-Mode性能分析任务名称，进入任务详情界面。

- 步骤三：在Top10函数区域单击某个热点函数，查看该函数的代码映射情况。

- 查看硬件事件类型、该事件的求和值、函数所在的源文件。
- 查看函数源代码。



NEW_01

函数分析任务 new_analysis_01 x LCPUTest2000::main x

事件事件 CPU CYCLES 总数 6409308878 文件名 /root/chenxi/CPUTest.java

源代码行号	源代码	数量(占比)
2	public static void main(String[] args) {	0 (0)
3	int busyTime = 10;	0 (0)
4	int idleTime = busyTime;	0 (0)
5	long startTime = 0;	0 (0)
6	long count = 0;	0 (0)
7	while (true) {	0 (0)

汇编指令地址	代码行号	汇编指令	数量(占比)
无数据			

代码流

无数据



查看所有函数分析结果

- 通过分析不同函数对应top-down模型的各指标值，按选定指标排序查看其TopN热点函数，可知针对某些硬件指标哪些函数是主要瓶颈。
 - 步骤一：单击已创建工程下的“函数分析任务”。
 - 步骤二：单击已创建的Java Mixed-Mode性能分析任务名称，进入任务详情界面。
 - 步骤三：单击“函数”页签，查看所有函数的各指标统计信息。



testxxx

函数分析任务 test_001 x

配置信息 | 采集日志 | 总览 | **函数** | 火焰图

Function/Callstack View

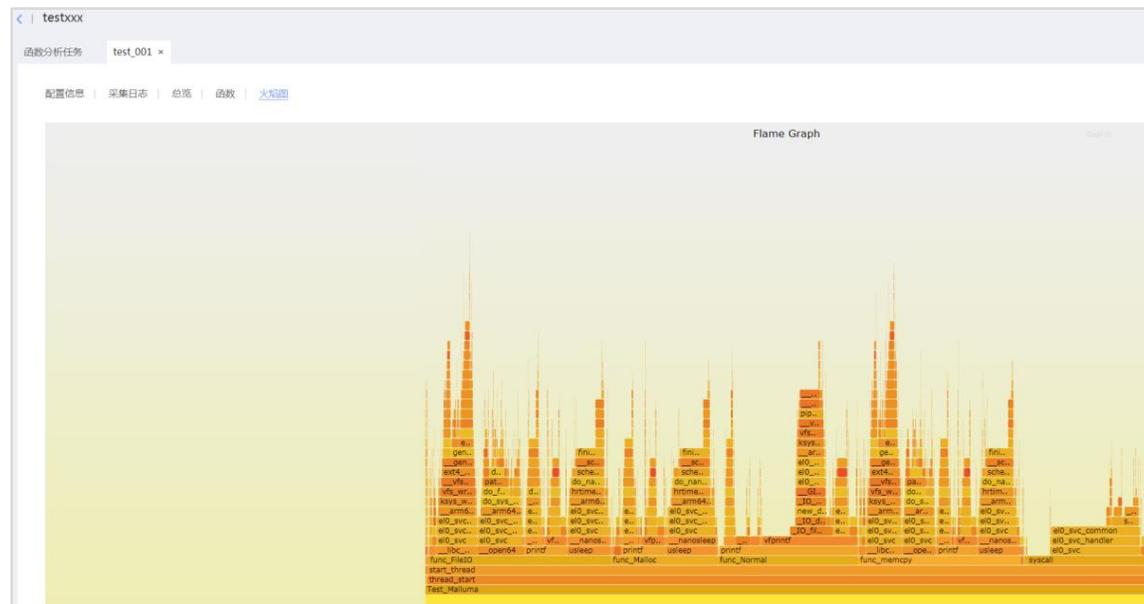
Function/Callstack	执行时间(秒)	时间间隔	指令数	IPC	模块
▼ Total	10.44266175	20885323500	1387511744	0.664347738736247	
> finish_task_switch	1.263357605	2526715210	188409902	0.7456713018322314	/proc/kcore
> ei0_svc_common	1.220340338	2440680676	148068780	0.6066700222442373	/proc/kcore
> djo_notify_resume	1.0606716155	2121343231	168427612	0.7939668109276372	/proc/kcore
> vforintf	0.572096014	1144192028	85844988	0.7502673143952372	/usr/lib64/libc-2.28.so
> syscall	0.418273633	836547266	53135022	0.6351705893926143	/usr/lib64/libc-2.28.so
> try_to_wake_up	0.37378186	747563720	30708097	0.4107756459877427	/proc/kcore
> _softirqentry_text_start	0.3645892305	729178461	79926246	1.0961136439821417	/proc/kcore
> _wake_up_common_lock	0.358902695	717805390	19127073	0.2664659985347839	/proc/kcore
> _audit_syscall_exit	0.254558665	509117330	30124221	0.5916950617257519	/proc/kcore



查看函数火焰图

- 通过查看采样时间热点函数的火焰图，帮助用户直观且迅速找到热点代码路径。

- 步骤一：单击已创建工程下的“函数分析任务”。
- 步骤二：单击已创建的Java Mixed-Mode性能分析任务名称，进入任务详情界面。
- 步骤三：单击“火焰图”页签，查看函数的火焰图。
- 步骤四：火焰图操作技巧。
 - 单击某个函数，可以放大显示该函数的调用栈。
 - 单击“Reset Zoom”即可恢复原火焰图。
 - 在火焰图右上角“Search”区域输入函数名称，可以快速定位到该函数。
 - 单击“Reset Search”即可恢复原火焰图。





思考题

1. 华为鲲鹏性能优化工具支持从哪些维度分析应用的性能瓶颈？（ ）
 - A. C/C++
 - B. Java Mixed-Mode
 - C. Locks and Waits
 - D. LLC&DDR
2. 华为鲲鹏性能优化工具能够提供（ ）方面的性能分析结果。
 - A. 分析Top热点函数
 - B. 分析函数火焰图
 - C. 分析热点函数代码映射
 - D. 分析不同函数对应top-down模型的各项指标值



本节小结

- 本节主要介绍了华为鲲鹏性能优化工具的定位、功能特性、应用场景、逻辑架构、访问方式和业务流程，以及如何使用该工具对C/C++类程序和Java Mixed-Mode类程序进行性能分析与优化的方法。



目录

1. 数据库性能测试
2. 大数据性能测试
3. HPC性能测试
4. 性能调优分析工具
5. 性能调优案例
 - NUMA优化
 - 线程并发数调整



本节概述和学习目标

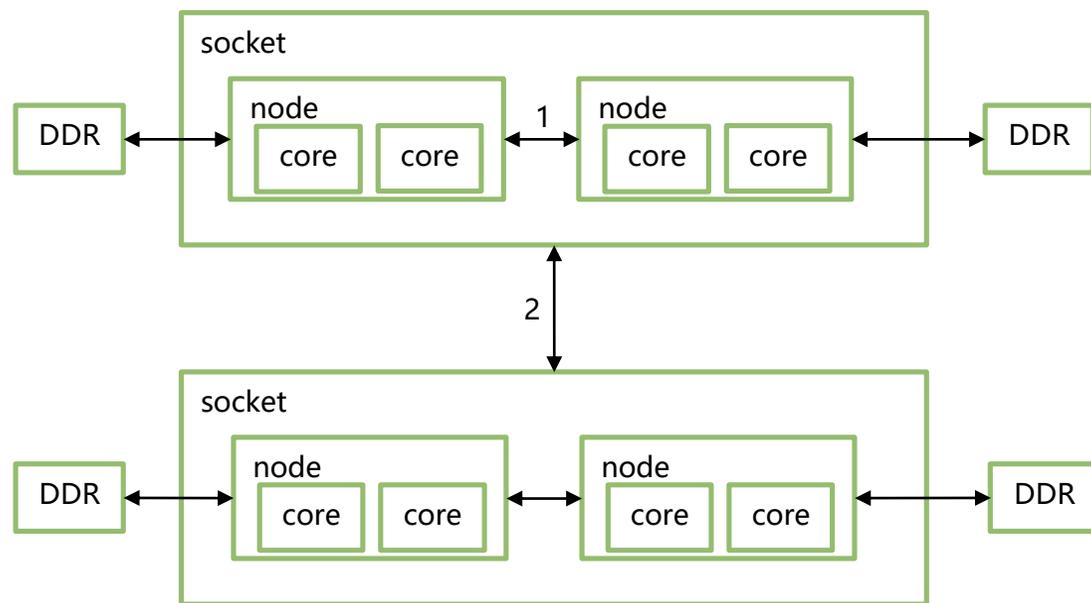
- 本节主要介绍了几个常用的性能调优案例，包括多CPU下的NUMA优化以及线程并发数的调整等。
- 学完本节内容后，您将能够：
 - 掌握NUMA优化的方法
 - 掌握线程并发数优化的方法



NUMA

- 物理上，一个DDR只挂载在一个node上，其它node要访问这个node上的DDR需要通过片内总线（如图中的1）或片间总线（如图中的2）进行通信，内存访问延迟从高到低为：

跨Socket > 跨NUMA不跨Socket > NUMA内





NUMA优化

- 基于NUMA的特点，为了更好地发挥系统性能，程序要**避免跨NUMA访问内存**：
 - 在网络中断的CPU占用高时，可以通过设置网卡中断的CPU亲和性，防止中断被跨NUMA处理：

```
echo $cpuMask > /proc/irq/$irq/smp_affinity_list
```
 - 通过numactl启动程序，如下面的启动命令表示启动test程序，且只能在core 28到core31运行（-C控制）：

```
numactl -C 28-31 ./test
```
 - 在C/C++代码中通过sched_setaffinity函数来设置线程亲和性；
 - 很多开源软件已经支持在自带的配置文件中修改线程的亲和性，如KVM开源软件，通过virsh edit \$vmname 修改cpuset等参数来设置线程的亲和性。
- 效果：在测试ceph软件的4K随机写性能时，通过设置线程的亲和性，性能提升了**32%**。



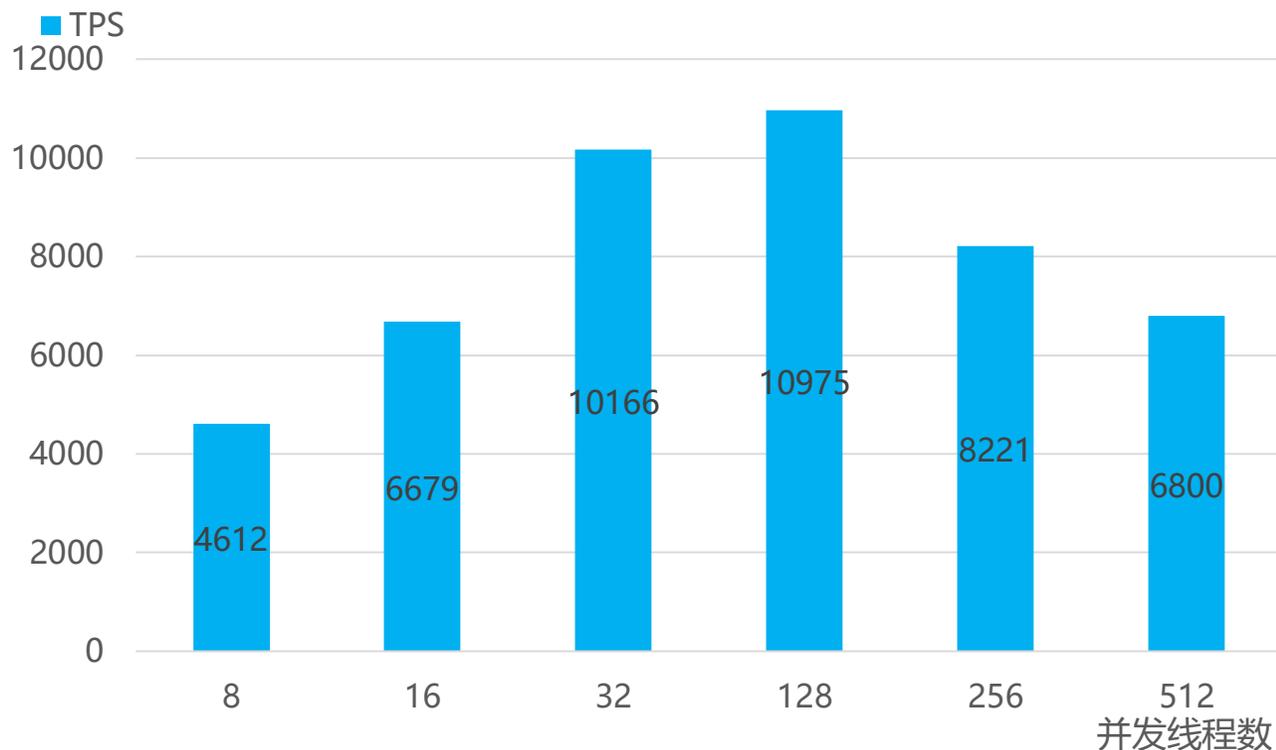
目录

1. 数据库性能测试
2. 大数据性能测试
3. HPC性能测试
4. 性能调优分析工具
5. 性能调优案例
 - NUMA优化
 - 线程并发数调整



线程并发数

- 应用程序从单线程变为多线程后，CPU和内存资源得到充分利用，性能会得到提升。但是超过一定限制后，因为资源的争抢比较严重，性能会下降。





线程并发数调整

- 对于线程并发数的调整，不同的软件有不同的配置方法，需要根据代码实现来修改。比如以下两个常用的开源软件的修改方法：
 - MySQL可以通过`innodb_thread_concurrency`参数，设置线程的并发数。
 - Nginx可以通过`worker_processes`参数，设置进程的并发数。
- 调整完成后，可以通过压力测试，测试出最大并发数，并观察性能是否有提升，从而找到最优配置。



思考题

1. NUMA架构下, 什么原因导致了CPU core访问不同的DDR的性能不一样?



本节小结

- 本节主要介绍了几个常用的性能优化方法，包括NUMA的优化以及线程并发数的调整。



本章总结

- 本章主要介绍了在数据库、大数据和HPC场景下进行性能测试的方法，介绍了华为鲲鹏性能优化工具Kunpeng Tuning Kit对C/C++类程序和Java Mixed-Mode类程序进行性能分析的方法，以及对NUMA和线程并发数进行性能调优的案例。



谢谢

www.huawei.com



应用部署和发布



前言

- 本章讲述了如何在鲲鹏计算平台上部署和发布应用，主要包括以下几方面的内容：
 - 如何在鲲鹏云服务器上搭建开发环境
 - 如何制作RPM包
 - 如何通过云服务器创建私有镜像
 - 如何梳理鲲鹏计算平台上部署和发布应用的常见问题。



目标

- 学完本课程后，您将能够：
 - 了解云服务器主要参数，以及掌握云服务器的购买流程
 - 掌握Linora交叉编译器的安装过程
 - 掌握RPM的打包原理，以及RPM包的制作过程
 - 掌握镜像的制作过程，以及使用镜像创建云服务器的过程
 - 了解应用部署常见问题和解决思路



目录

- 1. 应用部署与发布概述**
2. 鲲鹏开发环境搭建
3. 软件包制作
4. 镜像制作
5. 常见问题及解决方案



应用部署与发布概述

- 为了实现应用程序运行在鲲鹏计算平台上，我们需要具备两个条件：首先，具备鲲鹏开发环境，其次，具备应用程序的部署能力。
- 当前鲲鹏开发环境搭建有三种方式：

鲲鹏云服务器



优势：获取简单，规格丰富，按需计费

Taishan物理服务器



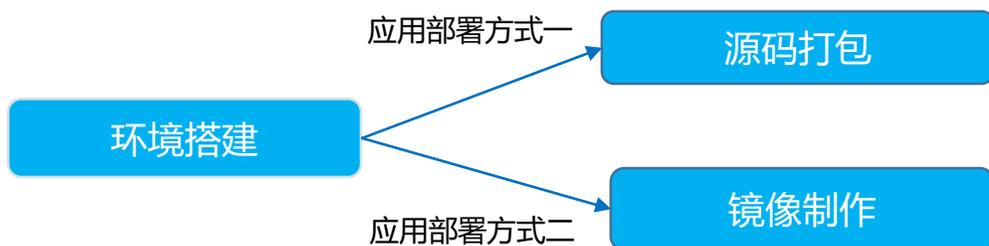
优势：物理服务器、性能强悍，自建机房使用

X86交叉环境编译



优势：如果基于已有环境搭建，无需额外成本

- 在鲲鹏平台上，应用的快速部署可以通过打包源码或者制作主机镜像的方式来实现。





目录

1. 应用部署与发布概述
- 2. 鲲鹏开发环境搭建**
 - 云服务器购买
 - 交叉环境搭建
3. 软件包制作
4. 镜像制作
5. 常见问题及解决方案



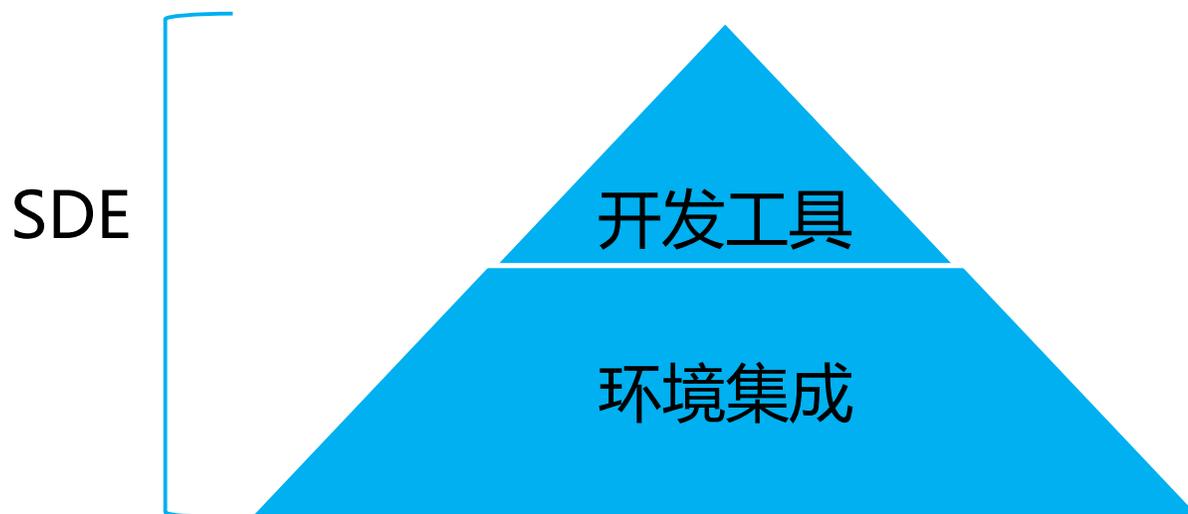
本节概述和学习目标

- 本节我们先一起学习一下开发环境搭建的过程。搭建开发环境，首先必须具备运行软件开发相关工具的平台，这里使用的是云服务器。其次，需要具备对应用进行代码转换的工具，这里使用的是Linaro交叉编译工具。
- 学完本节后，您将能够：
 - 了解云服务器的常见指标
 - 掌握云服务器的创建流程
 - 理解交叉编译的基本原理
 - 掌握x86服务器上交叉编译器安装
 - 通过一个简单例子测试交叉编译是否成功。



开发环境概述

- 软件开发环境(Software Development Environment, SDE)是指在基本硬件和宿主软件的基础上, 为支持系统软件和应用软件的工程化开发和维护而使用的一组软件, 简称SDE。它由软件工具和环境集成机制构成, 前者用以支持软件开发的相关过程、活动和任务, 后者为工具集成和软件的开发、维护及管理提供统一的支持。





开发环境搭建流程

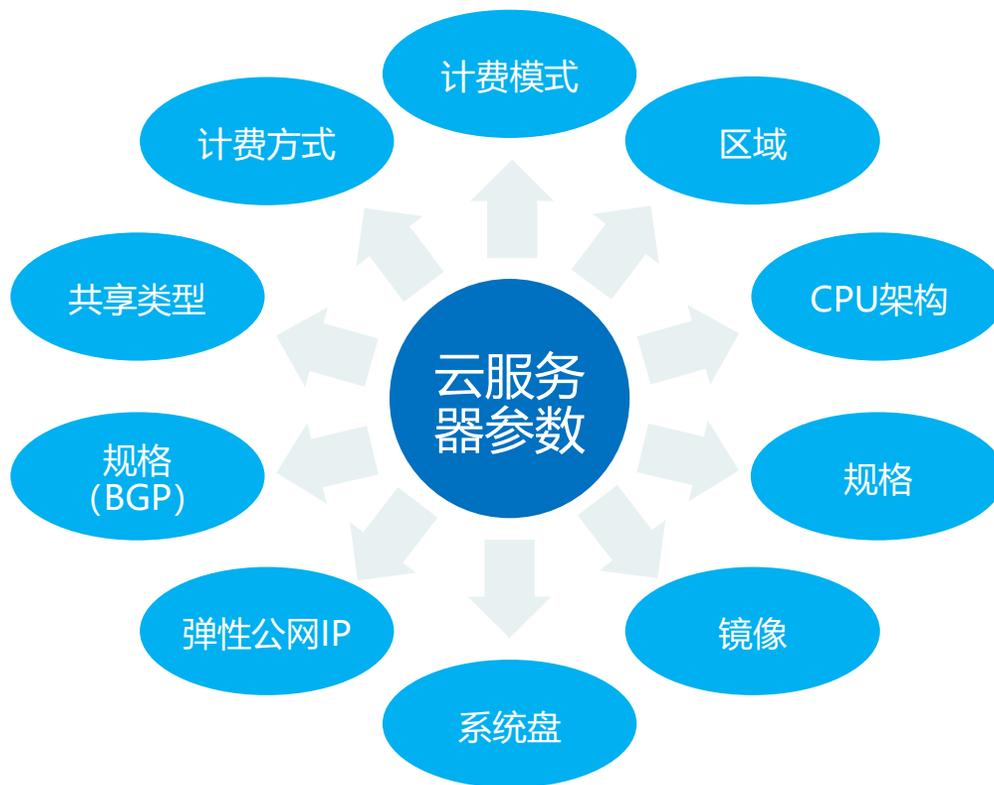
- 本节主要展示了通过在云服务器上部署Linaro交叉编译工具，来部署软件开发的环境。整个开发环境搭建可分为两大部分：云服务器的购买，Linaro交叉工具的部署。





云服务器的主要参数指标

- 云服务器，是一种简单高效、安全可靠、处理能力可弹性伸缩的计算服务。创建云服务器时往往包含以下常见的指标：





计费方式 & 区域

计费方式	内容
包年包月	不管开不开机，固定收费；
按需计费	按需计费是根据你服务器开机付费，关机不收费，可按小时购买

区域	内容
华北-北京一、华北-北京四、华东-上海一、华东-上海二、西南-贵阳一、华南-广州、 亚太-香港、亚太-曼谷、亚太-新加坡、 非洲-约翰内斯堡 ...	区域是指提供具体资源的数据中心位置



CPU架构

CPU架构	内容
X86	<p>x86 CPU架构采用复杂指令集（CISC） CISC指令集的每个小指令可以执行一些较低阶的硬件操作，指令数目多而且复杂，每条指令的长度并不相同。由于指令执行较为复杂所以每条指令花费的时间较长。</p>
ARM	<p>鲲鹏 CPU架构采用RISC精简指令集（RISC） RISC是一种执行较少类型计算机指令的微处理器，它能够以更快的速度执行操作，使计算机的结构更加简单合理地提高运行速度，相对于X86 CPU架构具有更加均衡的性能功耗比。鲲鹏的优势是高密度低功耗，可以提供更高的性价比。</p>



云服务器机型规格

参数	应用场景
通用计算型	可适用于Web应用服务器、轻负载应用、开发测试环境。中小型数据库系统、缓存、搜索集群。
通用入门型	适合平时都保持很低的CPU利用率，但偶尔需要瞬时CPU占用超高的场景
通用计算增强型	相比通用计算型，通用计算增强型弹性云服务器提供了更大规格的CPU和内存组合，可选范围更广
内存优化型	场景特点：内存要求高，同时要求内存优化。适用场景：大数据分析，如广告精准营销、电商、车联网等大数据分析场景
超大内存型	超大内存型弹性云服务器内存要求高，数据量大并且数据访问量大，同时要求快速的数据交换和处理以及低延迟的存储资源。提供超大内存，且有很高的计算、存储、网络能力
GPU加速型	提供强大的浮点计算能力，从容应对高实时、高并发的海量计算场景。
高性能计算型	可适用于基因工程、游戏动画、生物制药的计算和存储系统；渲染农场、动漫影视基地等公共渲染平台，以及影视等渲染平台；高性能前端集群、Web 服务器、高性能科学和工程应用、广告服务、视频编码和分布式分析等等
磁盘增强型	分为D1，D2型
FPGA加速型	提供FPGA开发和使用的工具及环境，让用户方便地开发FPGA加速器和部署基于FPGA加速的业务
GPU加速型	提供强大的浮点计算能力，从容应对高实时、高并发的海量计算场景



镜像

镜像类型	说明
公共镜像	常见的标准操作系统镜像，所有用户可见，包括操作系统以及预装的公共应用。 (公共镜像支持的操作系统请参考华为官网说明)
私有镜像	用户基于弹性云服务器或者云硬盘备份（系统盘备份）创建的个人镜像，仅用户自己可见。包含操作系统、预装的公共应用以及用户的私有应用。 私有镜像包括系统镜像和数据镜像，其中： (1) 系统镜像：包含用户运行业务所需的操作系统、应用程序的镜像。系统镜像可以用于创建云服务器，迁移用户业务到云。 (2) 数据镜像：只包含用户业务数据的镜像。数据镜像可以用于创建云硬盘，将用户的业务数据迁移到云上。
共享镜像	由其他用户共享的私有镜像。
市场镜像	提供预装操作系统、应用环境和各类软件的优质第三方镜像。无需配置，可一键部署，满足建站、应用开发、可视化管理等个性化需求。



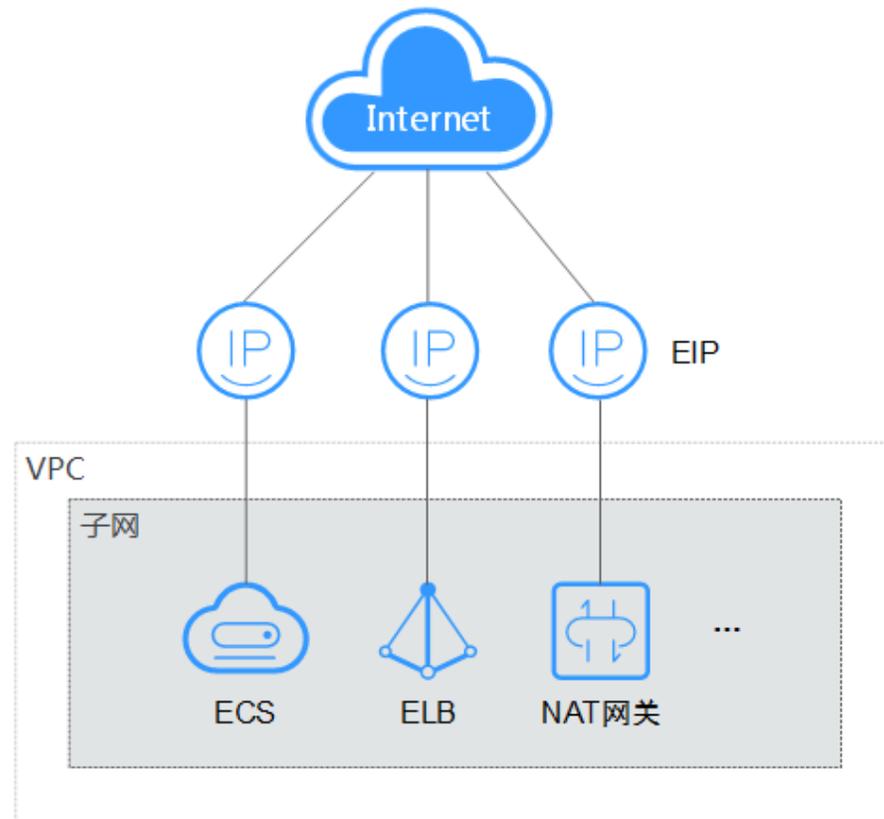
系统盘

系统盘类型	说明
普通IO	该类型云硬盘的最大IOPS为2200，适用于大容量、读写速率中等、事务性处理较少的应用场景，例如企业的日常办公应用或者小型测试等
高速高IO	该类型云硬盘的最大IOPS可达5000，最低读写时延为1 ms，适用于主流的高性能、高可靠应用场景，例如企业应用、大型开发测试以及Web服务器日志等。
超高IO	该类型云硬盘的最大IOPS可达33000，最低读写时延为1 ms，适用于超高IO，超大带宽的读写密集型应用场景，例如高性能计算应用场景，用来部署分布式文件系统，或者I/O密集型应用场景，用来部署各类NoSQL/关系型数据库。
超高IO (时延优化)	该类型的云硬盘提供低至1 ms的读写时延和高达1 GB/s的吞吐量，可运行企业核心业务，如SAP HANA。



弹性公网IP

- 弹性公网IP（Elastic IP，简称EIP）提供独立的公网IP资源，包括公网IP地址与公网出口带宽服务。可以与弹性云服务器、裸金属服务器、虚拟IP、弹性负载均衡、NAT网关等资源灵活地绑定及解绑。拥有多种灵活的计费方式，可以满足各种业务场景的需要。
- 一个弹性公网IP只能绑定一个云资源使用。
- 如右图通过EIP访问公网





BGP类型

类型	说明
静态BGP	网络结构发生变化时，无法实时自动调整网络设置以保障用户体验。
全动态BGP	可以根据设定的寻路协议实时自动优化网络结构，以保持客户使用的网络持续稳定、高效。



共享带宽

- **宽带类型：独享和共享**

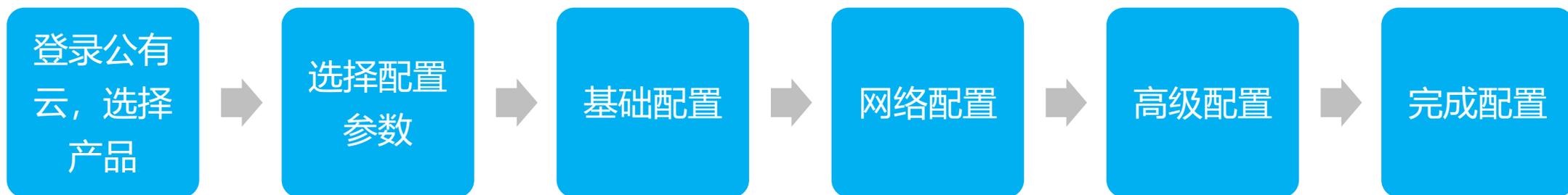
- 目前弹性IP的带宽支持独享带宽和共享带宽两种方式。
- 共享带宽可以实现多个弹性公网IP共同使用一条带宽。提供区域级别的带宽共享及复用能力，同一区域下的所有已绑定弹性公网IP的弹性云服务器、裸金属服务器、弹性负载均衡等实例共用一条带宽资源。

- **共享宽带优点**

- **节省带宽使用成本：**提供区域级别的带宽复用共享能力，节省带宽使用的运营及运维成本。
- **操作灵活：**不区分弹性公网IP类型及绑定实例类型，随时从共享带宽中增加或移出按需计费的弹性公网IP。
- **计费方式灵活：**提供包年包月、按需计费两种计费模式。



云服务器创建流程

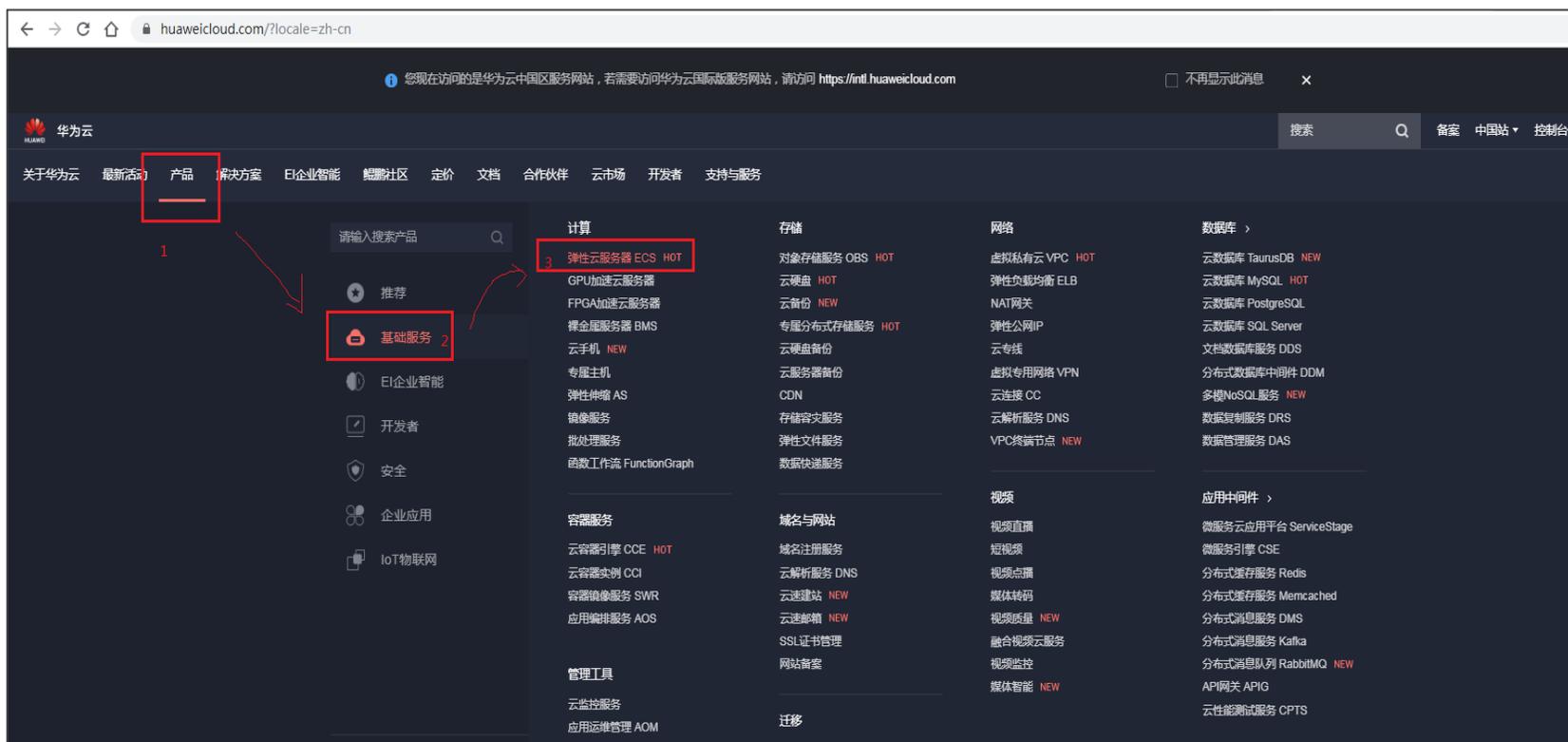




登录公有云，选择产品

这里以购买弹性云服务器ECS为例子

- 步骤一：登录公有云，按照如下步骤进入购买页面





选择配置参数

- 步骤二：根据实际参数配置选择购买。选择后，选择“去购买”进入下一步，对需要购买的云服务器进行配置。

产品价格详情 | 价格计算器 | 价格计算清单 ①: ¥0.00

弹性云服务器 ECS

计费模式: 包年包月 按需计费

区域: 华北-北京一 华北-北京四 华东-上海一 华东-上海二 西南-贵阳 华南-广州 亚太-香港 亚太-曼谷 亚太-新加坡 非洲-约翰内斯堡

CPU架构: X86计算 鲲鹏计算

X86采用复杂指令集 (CISC), ARM采用精简指令集 (RISC)。

规格: 通用计算型 通用入门型 通用计算增强型 内存优化型 超大内存型 GPU加速型

超高I/O型

S6 Sn3

1核 2核 4核 8核

4GB 8GB

选择的规格为 **s6.large.2 | 2核 | 4GB**

镜像: |

系统盘: GB

您还可以挂载 23 块磁盘 (云硬盘)

弹性公网IP:

规格:

共享类型:

计费方式:

带宽: Mbit/s

华为云按需EIP产品在绑定ECS、BMS、ELB、NAT产品后, 免收EIP保有费用
共享带宽: 一个带宽中可以加入多个弹性公网IP地址, 多个弹性公网IP地址共用一个带宽。

购买时长: 小时

购买数量: 台

配置费用 **¥ 1.21**



基础配置

- 步骤三：基础配置，依照需求选择配置参数。

弹性云服务器 < 返回云服务器列表 自定义购买 快速购买 New!

1 基础配置 2 网络配置 3 高级配置 4 确认配置

计费模式 包年/包月 按需计费

区域

不同区域的云服务产品之间内网互不相通；请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。

可用区 随机分配 可用区1 可用区2 可用区3

CPU架构 X86计算 鲲鹏计算

规格 vCPUs 内存 规格名称

规格名称	vCPUs 内存	CPU	基准 / 最大带宽	内网收发包	规格参考价
<input checked="" type="radio"/> s6.large.2	2vCPUs 4GB	Intel Cascade Lake 2.6GHz	0.2/1.5 Gbit/s	150,000	¥0.36/小时
<input type="radio"/> s6.large.4	2vCPUs 8GB	Intel Cascade Lake 2.6GHz	0.2/1.5 Gbit/s	150,000	¥0.68/小时
<input type="radio"/> sn3.large.2	2vCPUs 4GB	Intel SkyLake 6161 2.2GHz	0.35/1.5 Gbit/s	150,000	¥0.38/小时

镜像 公共镜像 私有镜像 共享镜像 市场镜像

系统盘 GB IOPS上限1,440, IOPS突发上限5,000

[+ 增加一块数据盘](#) 您还可以挂载 23 块磁盘 (云硬盘)

购买量 台 配置费用 ¥0.3796/小时
参考价格，具体扣费请以账单为准。 [了解计费详情](#)

[下一步: 网络配置](#)



网络配置

- 步骤四：网络配置，设置适合的网络配置。

弹性云服务器 < 返回云服务器列表 自定义购买 快速购买 New!

基础配置 2 网络配置 3 高级配置

网络 vpc-444(192.168.0.0/16) subnet-3a2c(192.168.0.0/24) 自动分配IP地址 可用私有IP数量238个 ?

如需创建新的虚拟私有云, 您可前往控制台创建。

扩展网卡 + 增加一块网卡 您还可以增加1块网卡

安全组 sg-86c4 (入方向:TCP/3389, 22 | 出方向: ... x) 新建安全组 ?

入方向:TCP/3389, 22| 出方向: -

弹性公网IP 现在购买 使用已有 暂不购买 ?

规格 全动态BGP

带宽类型 独享带宽 共享带宽

计费方式 按带宽计费 按流量计费 ?

带宽 5 Mbit/s

1 100 200 300 500 1,000,000 5 Mbit/s



高级配置

- 步骤五：高级配置

此步骤需要配置云服务器名称，设置root密码，此步骤设置后，可通过此用户名和密码进行远程登录控制。

弹性云服务器 < 返回云服务器列表

基础配置 网络配置 3

云服务器名称 允许重名
购买多台云服务器时，名称自动按序增加4位数字后缀。例如：输入ecs，从ecs-0001开始命名；若已有ecs-0010，从ecs-0011开始命名。

登录方式 密码 密钥对 创建后设置

用户名

密码
请妥善保管密码，系统无法获取您设置的密码内容。

确认密码

高级选项 现在配置



完成配置

- 步骤六：完成配置后，同意协议并确认购买。

弹性云服务器 < 返回云服务器列表

基础配置 网络配置 高级配置 **4** 确认配置

配置

配置	基础配置				
计费模式	按需计费	地域	北京一	可用区	可用区3
规格	通用计算型 s3.small.1 1vCPUs 1GB	镜像	CentOS 7.6 64bit	系统盘	高IO,40GB

网络配置

虚拟私有云	vpc-ijx11(192.168.0.0/16)	安全组	kubernetes.io-default-sg	主网卡	subnet-ijx11(192.168.0.0/24)
弹性公网IP	静态BGP 计费方式: 按带宽计费 带宽: 5 Mbit/s				

高级配置

云服务器名称	ecs-ca54	登录方式	密码
--------	----------	------	----

企业项目

购买数量 [您还可以创建7,996台云服务器。一次最多可以创建500台云服务器。申请更多云服务器配额请单击\[申请扩大配额\]\(#\)。](#)

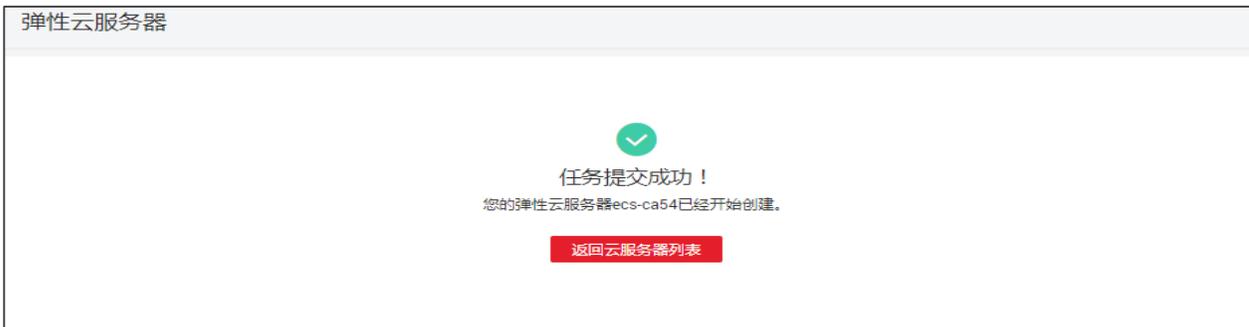
配置费用 **¥0.36**/小时 我已经阅读并同意 [《华为镜像免责声明》](#)

参考价格，具体扣费请以账单为准。 [了解计费详情](#)



完成购买，返回云服务列表

- 步骤七、完成购买，返回云服务列表。
- 在返回的服务列表里，可以通过远程登录验证是否购买成功了。



名称/ID	可用区	状态	规格/镜像	IP地址	计费模式	企业项目	操作
<input type="checkbox"/> ecs-ca54 6eeee9e6-eda0-4941-838e-d38a158b7272	可用区3	运行中	1vCPUs 1GB s3.small.1 CentOS 7.6 64bit	114.115.222.227 (弹性公网) 5 Mbit/s 192.168.0.9 (私有)	按需计费	default	远程登录 更多
<input type="checkbox"/> ecs-nwtest d945da9d-4797-4829-b3f5-219b6ac912db	可用区3	运行中	1vCPUs 1GB s3.small.1 CentOS 7.6 64bit	192.168.0.133 (私有)	按需计费	default	远程登录 更多



目录

1. 应用部署与发布概述
- 2. 鲲鹏开发环境搭建**
 - 云服务器购买
 - 交叉环境搭建
3. 软件包制作
4. 镜像制作
5. 常见问题及解决方案



编译

- 编译通常指利用编译程序从源语言编写的源程序产生目标程序的过程，或者动作。一般包括两种类型，本地编译和交叉编译。

本地编译

本地编译可以理解为，在当前编译平台下，编译出来的程序只能放到当前平台下运行。平时我们常见的软件开发，都是属于本地编译：

比如，我们在 x86 平台上，编写程序并编译成可执行程序。这种方式下，我们使用 x86 平台上的工具，开发针对 x86 平台本身的可执行程序，这个编译过程称为本地编译。

交叉编译

交叉编译可以理解为，在当前编译平台下，编译出来的程序能运行在体系结构不同的另一种目标平台上，但是编译平台本身却不能运行该程序：

比如，我们在 x86 平台上，编写程序并编译成能运行在 ARM 平台的程序，编译得到的程序在 x86 平台上是不能运行的，必须放到 ARM 平台上才能运行。



为什么会有交叉编译?

- Speed: 目标平台的运行速度往往比主机慢得多, 许多专用的嵌入式硬件被设计为低成本和低功耗, 没有太高的性能
- Capability: 整个编译过程是非常消耗资源的, 嵌入式系统往往没有足够的内存或磁盘空间
- Availability: 即使目标平台资源很充足, 可以本地编译, 但是第一个在目标平台上运行的本地编译器总需要通过交叉编译获得
- Flexibility: 一个完整的Linux编译环境需要很多支持包, 交叉编译使我们不需要花时间将各种支持包移植到目标板上



环境准备

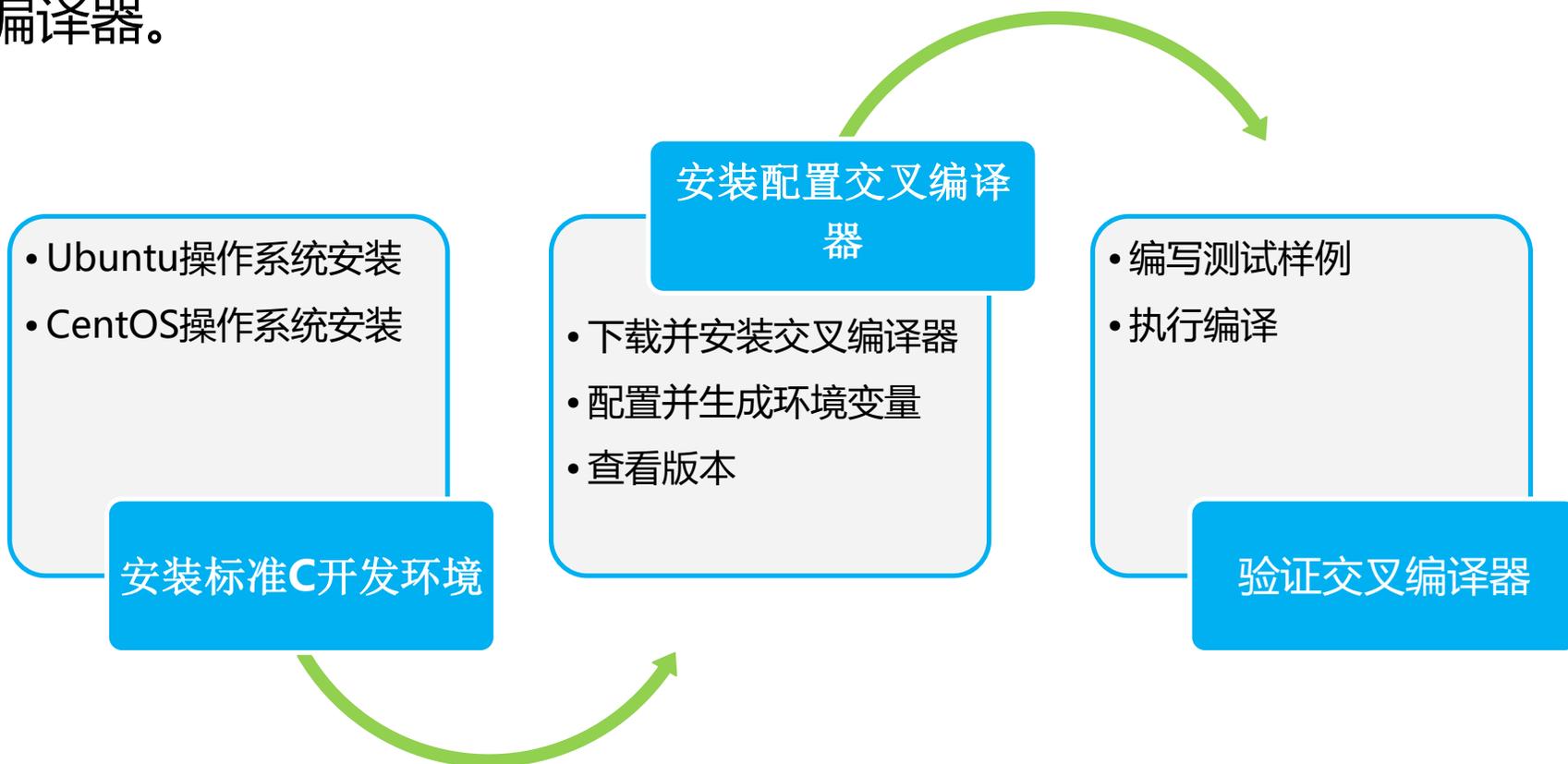
类别	操作系统版本	配置	数量	作用
鲲鹏测试服务器	CentOS7.6	kc1.large.2 2核 4GB	1	Arm平台，用于验证x86编译的可执行程序
x86编译服务器	CentOS7.6	s6.large.2 2核 4GB	1	用户搭建交叉编译环境，在此环境可编译处Arm运行的可执行程序
x86编译服务器	Ubuntu18.04	s6.large.2 2核 4GB	1	用户搭建交叉编译环境，在此环境可编译处Arm运行的可执行程序
交叉编译器	arm-linux-gcc	5.5.0	1	交叉编译的包 ，本实验在CentOS7.6 x86编译服务器和Ubuntu18.04 x86编译服务器两个环境上搭建

说明：X86编译服务器的操作系统版本二选一即可，本次课程对应实验是使用的CentOS 7.6.



环境搭建和验证流程

- 编译环境的搭建，通常包含以下三个步骤：安装基础环境，安装配置编译器，和验证交叉编译器。





安装配置交叉编译环境 (1)

- 步骤一：安装标准的C开发环境

ubuntu执行以下命令：

```
# apt-get install build-essential
```

CentOS执行以下命令：

```
# yum groupinstall Development Tools
```

- 步骤二：在/usr/local下建立名为ARM-toolchain的文件夹

```
# mkdir /usr/local/ARM-toolchain
```

- 步骤三：下载gcc-linaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu.tar.xz

```
# cd /usr/local/ARM-toolchain  
# wget https://releases.linaro.org/components/toolchain/binaries/latest-5/aarch64-linux-gnu/gcc-linaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu.tar.xz
```



安装配置交叉编译环境 (2)

- 步骤四: 解压

```
# tar -xvf gcc-linaro-5.5.0-2017.10-x86_64_aarch64-linux-gnu.tar.xz
```

- 步骤五: 配置环境变量

- 修改配置文件, 在配置文件的最后一行加入路径配置:

Ubuntu:

```
# vim /etc/bash.bashrc
```

CentOS:

```
# vim /etc/profile
```

- 步骤六: 环境变量生效

Ubuntu:

```
# source /etc/bash.bashrc
```

CentOS:

```
# source /etc/profile
```



交叉编译器验证

- 步骤一、编写测试例子

这里编写测试例子hello.c, 代码如下:

- 步骤二、执行如下命令, 交叉编译

```
# aarch64-linux-gnu-gcc -o kp-hello hello.c
```

- 步骤三: 测试执行

在x86服务器上测试, 执行:

```
# ./kp-hello
```

此时会报错, 报错内容为: `-bash: ./kp-hello: cannot execute binary file`

将持续拷贝到ARM服务器上, 继续执行, 显示内容如下:

```
hello linux
```



思考题

1. 什么是交叉编译？为什么需要交叉编译？



本节小结

- 本节主要讲述了鲲鹏云服务器开发环境的搭建，包括如下：
 - 云服务器相关参数功能
 - 购买华为云服务器的流程
 - 交叉编译的基础知识
 - Linaro交叉环境搭建



目录

1. 应用部署与发布概述
2. 鲲鹏开发环境搭建
- 3. 软件包制作**
 - Linux环境下软件安装基础介绍
 - RPM介绍
 - Redis源码包制作成RPM包
4. 镜像制作
5. 软件包安装常见问题



本节概述和学习目标

- 上一小节主要讲述了如何部署软件开发环境，并配置交叉编译工具，本节主要讲解如何对应用进行打包。虽然现在CentOS操作系统早已经完成了对yum的支持，但是由于yum依然还是基于RPM机制的，所以本节介绍了RPM打包的原理，过程，优点和必要性。
- 学完本节后，您将能够：
 - 为什么要打包
 - 如何制作RPM打包
 - RPM打包原理
 - Redis源码包制作成RPM包



Linux安装软件

- Linux目前安装软件的方式有三种：源码安装，yum安装软件，RPM安装安装。
- 本节主要介绍RPM方式。

源码安装

- 可以自定义安装目录和一些配置文件
- 但需要事先调整编译参数

Yum安装软件

- 全自动安装，自动安装依赖
- 但缺乏自主性，软件的功能和存放的位置均已设置好

RPM安装

- 自主制作的RPM包能够实现全自动安装，且可自定义安装路径等配置
- 但需提前识别依赖并手动安装



目录

1. 应用部署与发布概述
2. 鲲鹏开发环境搭建
- 3. 软件包制作**
 - Linux环境下软件安装基础介绍
 - RPM介绍
 - Redis源码包制作成RPM包
4. 镜像制作
5. 软件包安装常见问题



RPM概述

- RPM (Redhat Package Manager) 是用于Redhat、CentOS、Fedora等Linux操作系统的常见软件包管理器。它允许分发已编译的软件，支持一键安装软件。
- 由于 RPM 是过预先编译并打包成为 RPM 文件格式后，再加以安装的一种方式，并且还能够进行数据库的记载，所以 RPM 有以下的优点：

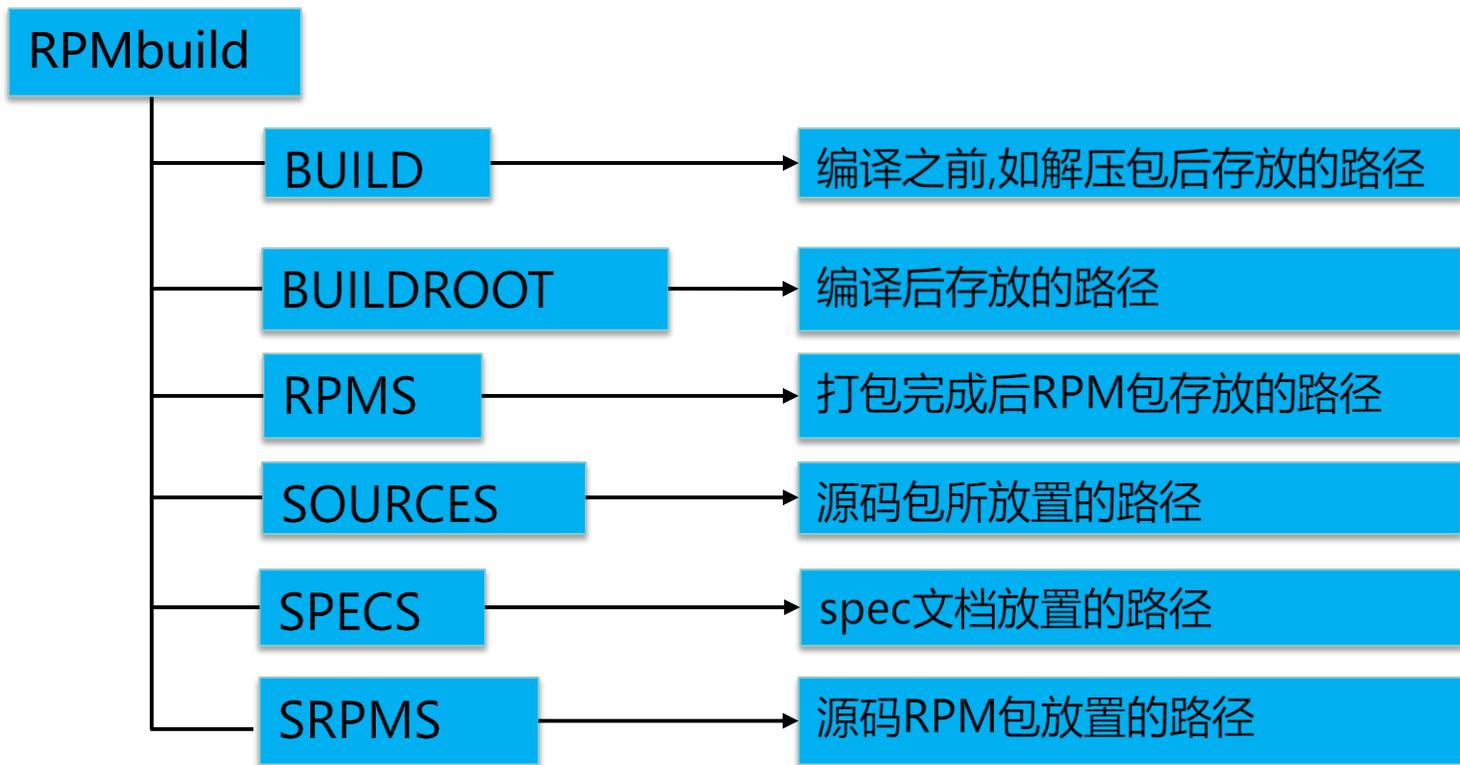




RPMbuild

RPMbuild是用来指示转换的源码编译成二进制文件的包, 如果想发布rpm格式的源码包或者是二进制包, 就要使用RPMbuild工具。

RPMbuild文件夹的目录结构如下:





RPM包制作流程

- 一个RPM的制作，通常包含以下5个步骤：

1. 将源代码放到SOURCES文件夹中



2. 在SPECS文件夹中配置spec文件



3. BUILD中完成源码的解压



4. 在BUILDROOT中完成软件的安装



5. 生成的RPM包放置到RPMS,RPM源码包放置到SRPMS下



目录

1. 应用部署与发布概述
2. 鲲鹏开发环境搭建
- 3. 软件包制作**
 - Linux环境下软件安装基础介绍
 - RPM介绍
 - Redis源码包制作成RPM包
4. 镜像制作
5. 软件包安装常见问题



环境准备

- Redis（Remote Dictionary Server）是一款内存高速缓存数据库。由于Redis具备极高的读写数据的效率，并且支持丰富的数据类型，所以被广泛使用在网站等其他领域。本小结以Redis源码包制作成RPM包为样例，详细讲解RPM包制作的过程。
- 在将Redis源码包制作成RPM包之前，需要做以下两个准备工作：
 - 准备一个Linux环境，比如CentOS,或者EulerOS。
 - 安装rpm-build包，因为RPM打包使用的是rpmbuild命令，这个命令来自rpm-build

```
# yum -y install rpm-build
```



Redis源码包制作成RPM包

- **1. 下载redis源码包**

- 请自行到官网搜索相应的下载链接。

```
# wget http://download.redis.io/releases/redis-4.0.9.tar.gz
```

- **2. 创建目录**

- RPM 打包的时候需要编译源码，还需要把编译好的配置文件,二进制命令文件放到合适的位置，创建rpmbuild目录：

```
# mkdir ~/rpmbuild
```

- 创建rpmbuild的子目录：

```
# mkdir -p ~/rpmbuild/{BUILD, BUILDROOT, RPMS, SOURCES, SPECS, SRPMS}
```



Redis源码包制作成RPM包

- **3. 编辑SPEC文件**
- 通过SPEC文件来配置Redis RPM包的安装过程:

```
# cd ~/rpmbuild/SPECS  
# vim redis.spec
```



Redis源码包制作成RPM包

4. SPEC配置文件如下:

```
Name: redis
Version: 4.0.9
Release: 1%{?dist}
Summary: This is a RedisDB
License: GPL
URL: https://ftp.postgressql.org
Source0: redis-4.0.9.tar.gz
BuildRequires: gcc
#Requires: zlib-devel,readline-devel

%description
RedisDB
%prep
%setup -q
%build
```

```
make %{?_smp_mflags}
%install
make install PREFIX=%{buildroot}%{_prefix}
install -p -D -m 644 %{name}.conf %{buildroot}%{_sysconfdir}/%{name}.conf
chmod 755 %{buildroot}%{_bindir}/%{name}-*
mkdir -p %{buildroot}%{_sbindir}
mv %{buildroot}%{_bindir}/%{name}-server %{buildroot}%{_sbindir}/%{name}-server
%clean
rm -rf %{buildroot}
%files
%defattr(-,root,root,-)
%{_bindir}/%{name}-*
%{_sbindir}/%{name}-*
%config(noreplace)%{_sysconfdir}/%{name}.conf
%changelog
```



SPEC配置文件参数详解

参数	详解
Name	软件包的名称, 后面可使用%{name}的方式引用
Version	软件包版本号, 后面可使用%{version}引用
Release	软件包的发行号, 后面可使用%{release}引用
Summary	简要说明
License	软件授权方式, 如GPL (自由软件) 或GPLv2,BSD
URL	软件的主页
Source0	源代码包的名字
%prep	预处理用来执行一些解开源程序包的命令, 为下一步的编译安装作准备
%setup	解压源代码, 将当前目录改为源代码解压之后产生的目录
%build	编译软件包所要执行的命令
%install	安装软件包时将执行命令
%clean	%clean: 清理 (作用同rm-rf \$RPM_BUILD_ROOT)
%files	软件包所包含的文件, 包括说明文档 (doc), 配置文件 (config) 及执行程序
%changelog	更新日志, 每次软件的更新内容可以记录在此到这里, 以便查用



构建和安装RPM包

- **5. 构建RPM包**

- 在SPECS目录下执行如下指令以生成二进制版本的RPM包:

```
# rpmbuild -bb redis.spec
```

- **6. 安装RPM包**

```
# RPM -ivh ~/RPMbuild/RPMS/ redis-4.0.9-1.el7.aarch64.RPM
```



备查

rpmbuild目录

默认位置	宏代码	名称	用途
~/rpmbuild/SPECS	_%specdir	spec文件目录	保存RPM包配置 (.spec) 文件
~/rpmbuild/SOURCES	_%sourcedir	源代码目录	保存源码包 (如tar包) 和所有patch补丁
~/rpmbuild/BUILD	_%buildir	构建目录	源码包被解压至此, 并在该目录的子目录完成编译
~/rpmbuild/BUILDROOT	_%buildrootdir	最终安装目录	保存%install阶段安装的文件
~/rpmbuild/RPMS	_%rpmdir	标准RPM包目录	生成保存二进制RPM包
~/rpmbuild/SRPMS	_%srcrpmdir	源代码RPM包目录	生成保存源码RPM包 (SRPM)



思考题

1. RPM打包使用的是什么命令，这个命令来自以下哪个包？（ ）
 - A. rpm , rpmbuild包
 - B. rpmbuild , rpm-build包
 - C. rpmbuild , rpmbuild包
 - D. rpm , rpm-build包
2. 下载的源码包放在哪个目录下？（ ）
 - A.BUILD
 - B.RPMS
 - C.SOURCES
 - D.SPEC



本节小结

- 本节主要讲述了RPMbuild 的建构和RMP包的制作流程。



目录

1. 应用部署与发布概述
2. 鲲鹏开发环境搭建
3. 软件包制作
- 4. 镜像制作**
 - 镜像创建
 - 镜像验证
5. 软件包安装常见问题



本节概述和学习目标

- 除了制作RPM包的方式快速部署应用之外，通过制作镜像包并在镜像市场中发布，也能快速的部署应用。本节主要以Linux系统盘镜像为例，讲述如何通过云服务器创建私有镜像，并通过私有镜像快速部署应用的过程。
- 学完本节后，您将能够：
 - 通过云服务器创建Linux系统盘镜像
 - 通过私有镜像创建云服务器



镜像服务的简要介绍

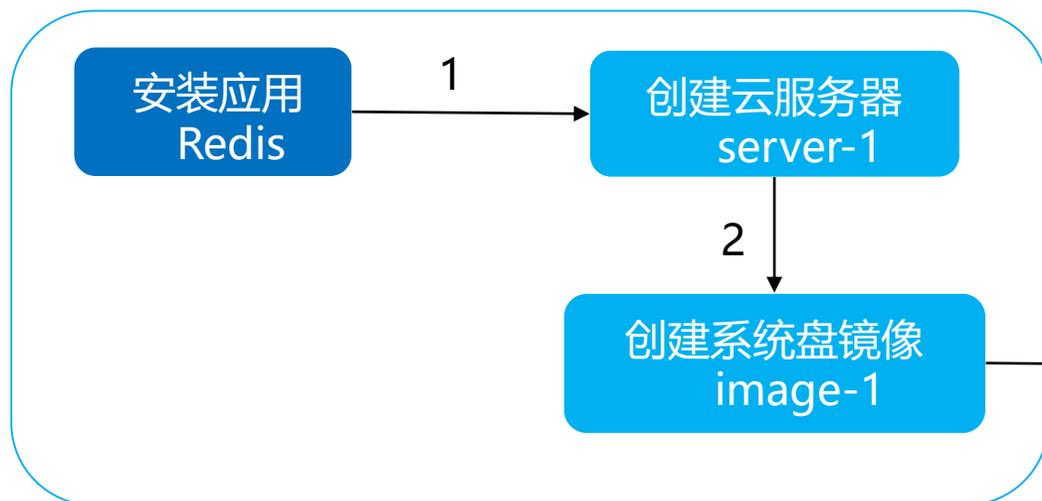
- 镜像是用于创建服务器或磁盘的模板。镜像分为公共镜像、私有镜像、共享镜像、市场镜像。公共镜像为系统默认提供的镜像，私有镜像为用户自己创建的镜像，共享镜像为其他用户共享的私有镜像。
- 镜像服务具有以下功能：
 - 提供常见的主流操作系统公共镜像（支持的操作系统类型请以控制台镜像服务页面的显示为准）。
 - 创建私有镜像。
 - 管理镜像。
 - 通过镜像创建云服务器。



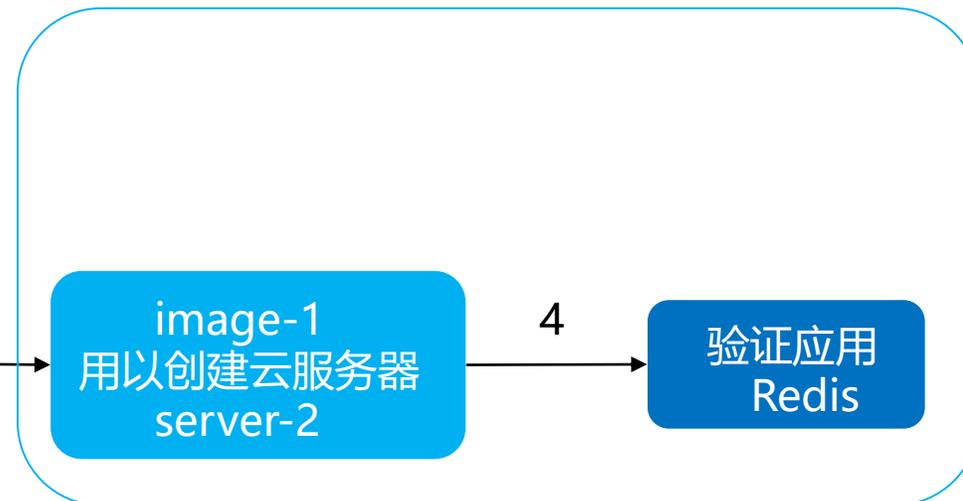
镜像制作流程

- 一个完整的镜像制作过程通常包含两个步骤：制作镜像，和验证镜像。
 - 镜像制作是将一个已经部署了应用的云服务器制作成模板。
 - 镜像验证是通过验证server-2上的Redis应用是否运行正常，以此来验证镜像是否创建成功，且数据一致。

镜像制作流程



镜像验证流程





通过云服务器创建Linux系统盘镜像（1）

- 前提条件

系统中依据存在一个包含一个系统盘、没有数据盘的云服务器 server-1。

操作步骤：

- 步骤一：登录华为云管理控制台。选择“基础服务 > 镜像服务”。在“镜像服务”列表页面，单击“创建私有镜像”。

镜像服务 ② + 创建私有镜像

目前镜像服务已进入商业化阶段，私有镜像会收取一定的存储费用，删除创建的镜像后将不再计费。详细计费标准可参考[镜像的计费标准](#)

公共镜像 私有镜像 共享镜像 到云市场发布镜像。了解发布镜像流程，请单击[帮助](#)

镜像支持云服务器快速发放，建议您优化不支持该功能的镜像。请在详情页面查看镜像是否支持快速发放。[了解更多](#)

您还可以创建95个私有镜像。

删除 共享 所有项目 所有镜像 所有操作系统 名称 Q 标签搜索 刷新 分享

名称	状态	操作系统类型	操作系统	镜像类型	磁盘容量 (GB)	加密	创建时间	企业项目	操作
<input type="checkbox"/> sxf-redis-image	正常	Linux	CentOS 7.6 64bit	ECS系统盘镜像(x86)	40	否	2019/09/20 15:58:44 GMT+08:00	default	申请服务器 修改 更多



通过云服务器创建Linux系统盘镜像（2）

- 步骤二：在“镜像类型和来源”页面，选择镜像的“创建方式”为“系统盘镜像”。“选择镜像源”默认为“云服务器”，从列表中选择相应的云服务器。如下图所示。

镜像类型和来源

* 区域

不同区域的资源之间内网不互通。请选择靠近您客户的区域，可以降低网络时延、提高访问速度。

* 创建方式 系统盘镜像 整机镜像 数据盘镜像

* 选择镜像源 云服务器 裸金属服务器 镜像文件

- 当前关机或开机状态的弹性云服务器可以用来创建私有镜像。
- 创建镜像前，请确保弹性云服务器已完成相关配置。[了解更多](#)
- 请勿在创建镜像过程中对所选择的弹性云服务器及其相关联资源进行其他操作。

所有状态

名称	操作系统	运行状态	私有IP地址	创建时间
<input type="checkbox"/> ecs-sxf-from-image-test	CentOS 7.6 64bit	<input checked="" type="radio"/> 运行中	192.168.1.172	2019/09/20 16:06:47 ...

6 总条数: 19 < 1 2 3 4 >



通过云服务器创建Linux系统盘镜像（3）

- 步骤三：在镜像配置信息页面，填写镜像的基本信息，例如，镜像名称 image-1 和描述信息。信息确认无误后，单击“立即申请”。
- 步骤四：根据界面提示，确认镜像参数。阅读并勾选协议，单击“提交申请”。提交完成后，根据界面提示，返回私有镜像列表
- 步骤五：查看镜像状态。
 - 镜像创建时间与镜像文件本身大小有关，大约需要20分钟，当镜像的状态为“正常”时，表示创建完成。
 - 创建成功后，在镜像列表中会显示 image-1 的详细信息。



通过云服务器创建Linux系统盘镜像（4）

说明：

- 1、在创建镜像过程中，请勿对所选择的云服务器及其相关联资源进行其他操作。
- 2、使用加密镜像创建的弹性云服务器为加密的弹性云服务器，加密的云服务器的密钥与加密镜像的密钥相同。
- 3、使用加密的弹性云服务器创建的镜像为加密镜像，该加密镜像的密钥与加密的云服务器的密钥相同。



目录

1. 应用部署与发布概述
2. 鲲鹏开发环境搭建
3. 软件包制作
- 4. 镜像制作**
 - 镜像创建
 - 镜像验证
5. 软件包安装常见问题



通过私有镜像创建云服务器（1）

- 通过上一小节创建的系统盘私有镜像 image-1 ，可以创建与原始云服务器 server-1 相同规格的云服务器 server-2，且用户运行业务所需的操作系统、应用软件都与 server-1 相同。
- 后续可以验证 server-2 上的应用软件是否运行正常，以此来验证系统盘镜像是否创建成功。



通过私有镜像创建云服务器（2）

- 步骤一：登录华为云，进入“弹性云服务器 ECS”页面，可以创建云服务器。
- 注意：
 - 在“基础配置”页面，“镜像”一栏选择“私有镜像”，下拉框中选择上一小节创建的系统盘镜像 image-1。其他配置可以选择与 server-1 相同的配置。



- 步骤二：配置完成后，“立即申请”创建云服务器server-2。
- 步骤三：验证Redis的功能



思考题

问题一：使用镜像创建云服务器的好处是什么？



本节小结

本小节主要讲解了如何通过云服务器创建私有镜像，以及使用私有镜像创建云服务器，验证镜像的功能。



目录

1. 应用部署与发布概述
2. 鲲鹏开发环境搭建
3. 软件包制作
4. 镜像制作
- 5. 软件包安装常见问题**



本节概述和学习目标

- 本节主要对鲲鹏平台部署常见问题进行讲解。
- 学完本节后，您将能够：
 - 了解鲲鹏平台部署常见问题和解决思路

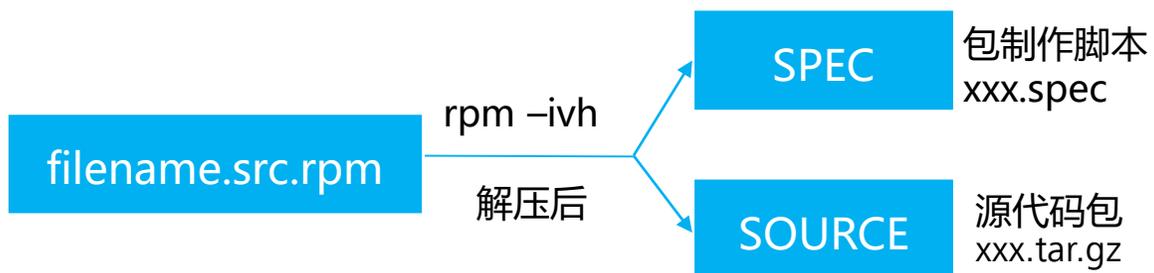


软件包制作常见问题

- 问题1：RPM和SRPM包（filename.rpm和filename.src.rpm）区别是什么？

文件格式	档名格式	是否可直接安装	内含文件类型	可否修改参数并编译
RPM	filename.rpm	可以	已编译的程序	不可
SRPM	filename.src.rpm	不可	未编译的源代码	可以

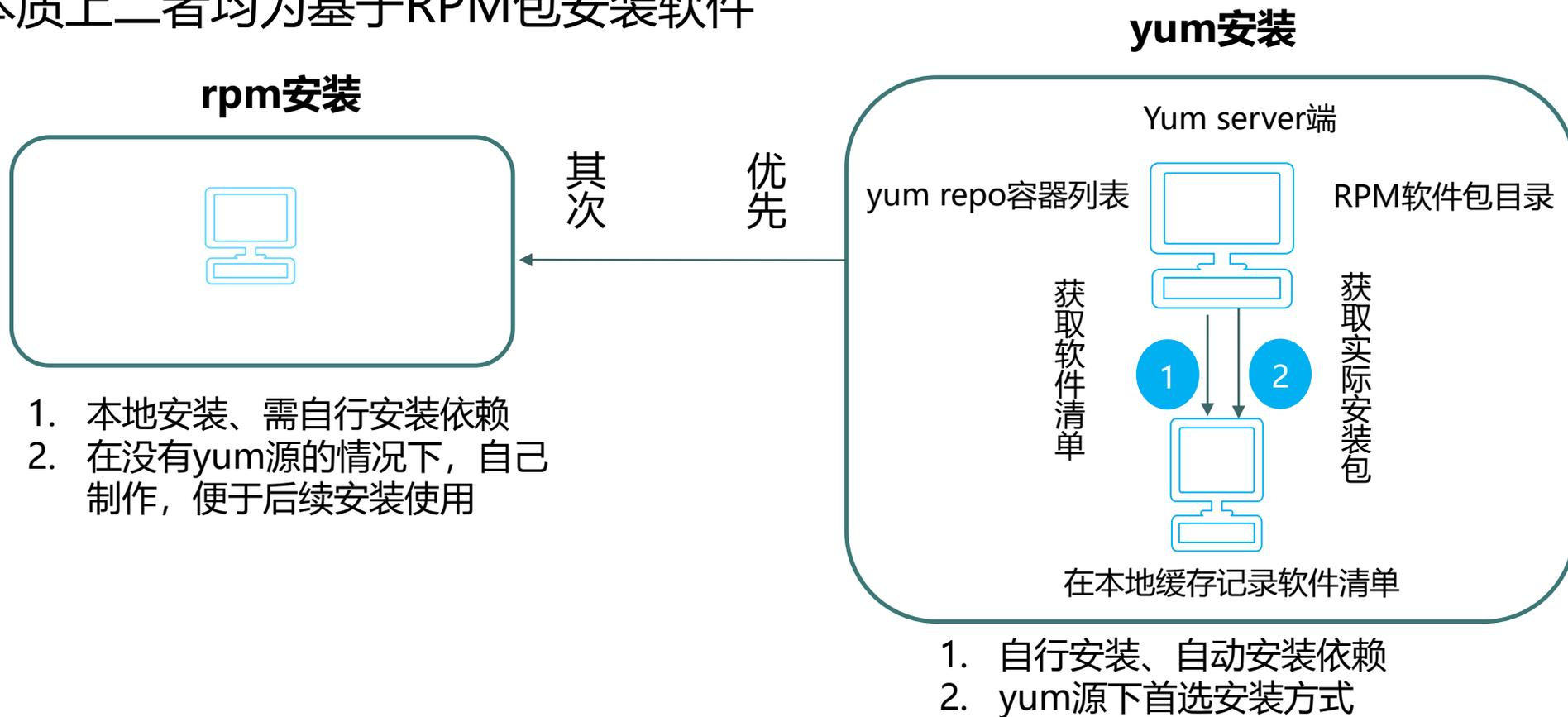
- SRPM的作用：用来二次制作适应于环境的RPM包，例如，在OS_1制作的RPM包，直接移植到OS_2上可能不可用，需要修改配置，那么可以使用SRPM包解压后，修改spec文件重新在OS_2上制作新的适应的RPM包





软件包安装常见问题

- 问题1：rpm方式安装与yum安装联系与区别？
- 本质上二者均为基于RPM包安装软件





镜像制作常见问题

- 问题1：如果我是嵌入式开发环境，我能否直接把我OS镜像运行到鲲鹏上？
- 分析：嵌入式系统大多是采用经过裁剪或者定制化的Linux/unix内核，对于该类型的OS，在鲲鹏计算平台需要定制化以来适配内核所需要的各类驱动等程序，当前鲲鹏上并不支持该类裁剪过的非通用*的OS。（通用*的OS：如CentOS/ubuntu/fedora/suse等。）
- 下表是华为鲲鹏云服务器支持的私有镜像操作系统类型列表（更新时间：2019-09-20）

操作系统类型	操作系统版本
CentOS	CentOS 7.6 64bit CentOS 7.5 64bit CentOS 7.4 64bit
EulerOS	EulerOS 2.8 64bit
Fedora	Fedora 29 64bit
Ubuntu	Ubuntu 18.04 server 64bit



镜像制作常见问题

- 问题2：如果我的弹性云服务器上使用了数据盘，还能制作镜像吗？
- 分析：可以，当前华为云镜像服务已支持如下三种模式制作整机镜像：
 - 使用云服务器制作整机镜像
 - 使用云服务器备份制作整机镜像
 - 使用云备份制作整机镜像
- 具体可以参考华为云IMS镜像服务：[制作整机镜像](#)



本章总结

- 通过本章的学习，应该掌握如何在云服务器上搭建软件开发环境，并且掌握如何进行交叉编译环境的搭建，掌握RPM包的制作，以及通过制作镜像包的方式快速发布应用。



谢谢

www.huawei.com



华为鲲鹏平台应用软件移植调优综合实验



前言

- 本章主要以PostgreSQL为例，介绍在华为鲲鹏平台上对应用软件进行移植调优的全流程综合实验，包括环境搭建、软件移植、性能测试、性能调优、软件打包等。



目标

- 学完本课程后，您将能够：
 - 掌握搭建华为鲲鹏平台实验环境的方法
 - 掌握在华为鲲鹏平台上进行软件移植的方法
 - 掌握在华为鲲鹏平台上进行软件性能测试和调优的方法
 - 掌握在华为鲲鹏平台上进行软件打包的方法



目录

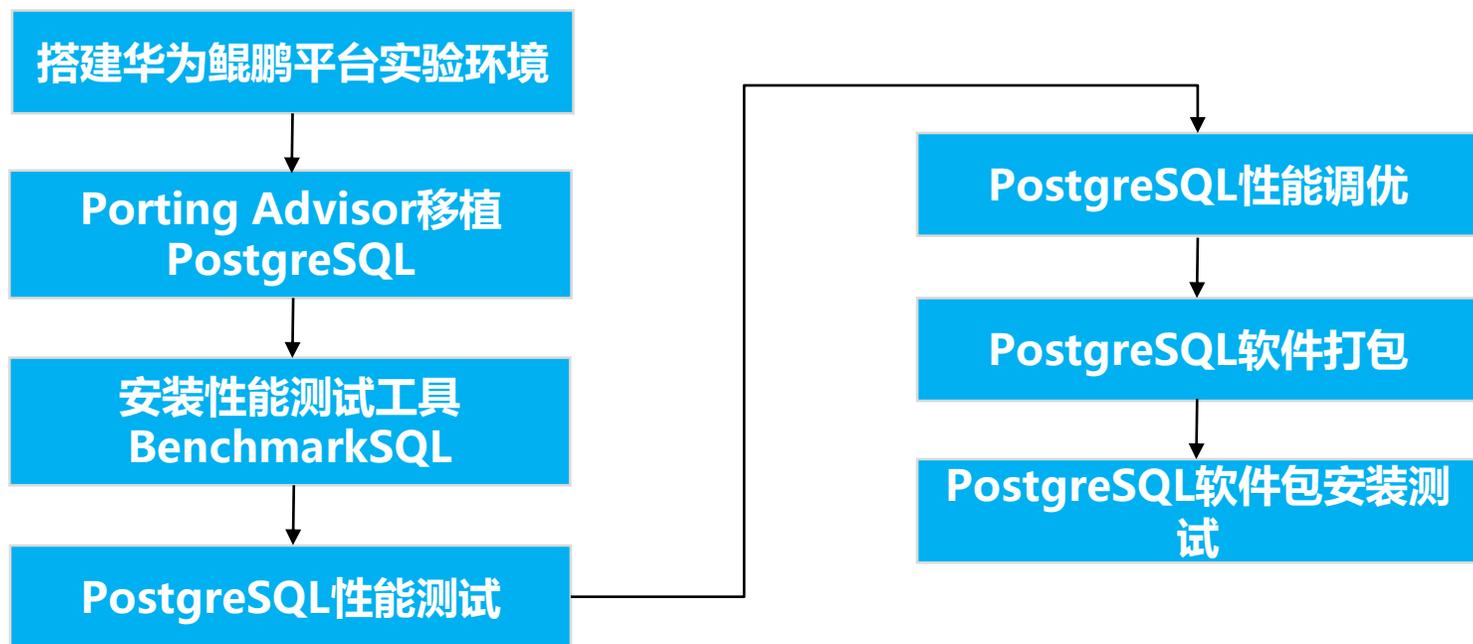
1. 实验流程介绍

- 实验流程概览
- 搭建华为鲲鹏平台实验环境
- 移植软件PostgreSQL
- 安装性能测试工具BenchmarkSQL
- PostgreSQL性能测试
- PostgreSQL性能调优
- PostgreSQL软件打包
- PostgreSQL软件包安装测试



实验流程概览

- 在华为鲲鹏平台上对应用软件进行移植调优的全流程包括环境搭建、软件移植、性能测试、性能调优、软件打包等，本综合实验将以PostgreSQL为例进行介绍。





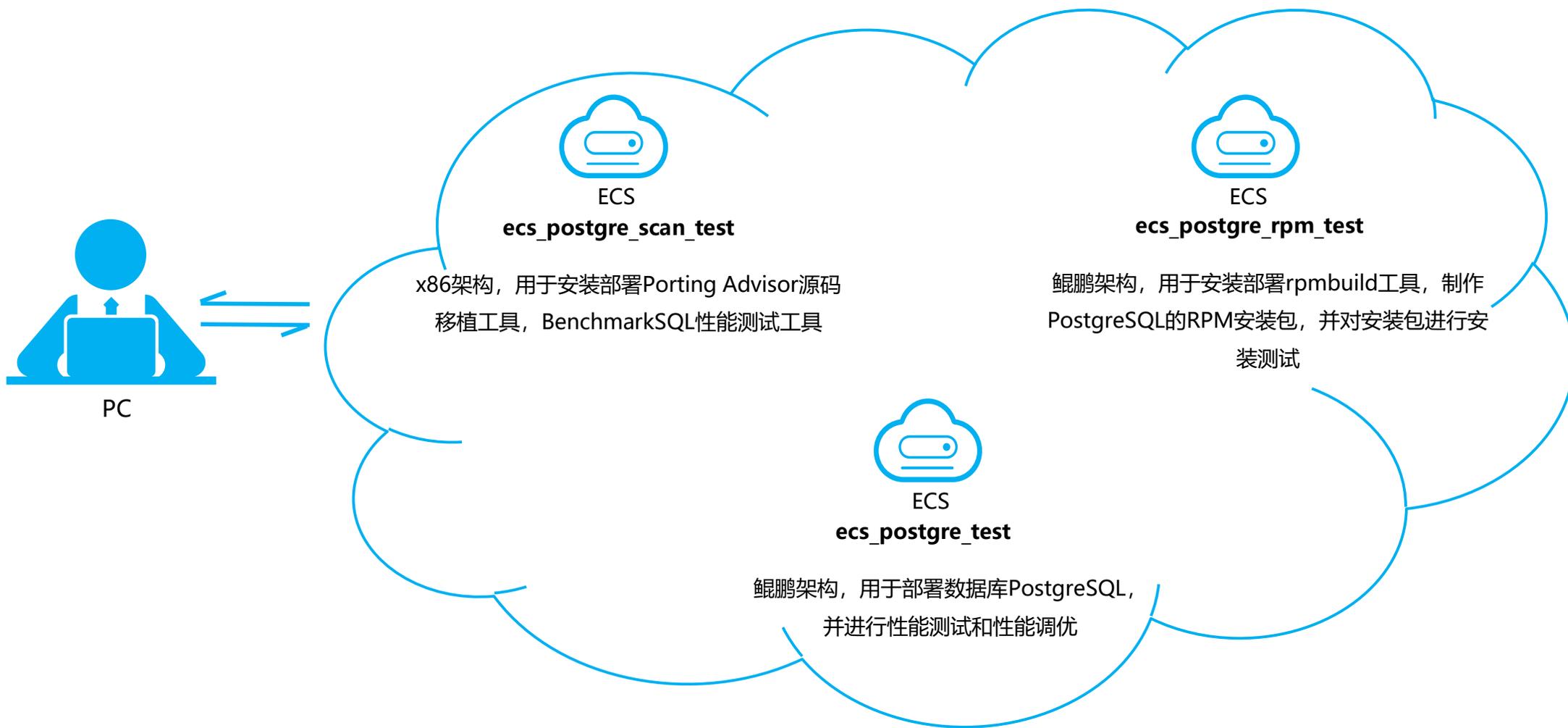
搭建华为鲲鹏平台实验环境

- 华为鲲鹏平台实验环境可以基于Taishan服务器和华为云ECS云服务器来搭建，基于便利性考虑，本实验采用的是华为云ECS云服务器。
- 通过华为云购买如下三台华为云ECS云服务器：

名称	云主机名称	配置	OS版本
x86云服务器	ecs_postgre_scan_test	c6.large.2 2vCPUs 4GB	CentOS 7.6 64bit
鲲鹏云服务器	ecs_postgre_test	kc1.2xlarge.2 8vCPUs 16GB	CentOS 7.6 64bit
鲲鹏云服务器	ecs_postgre_rpm_test	kc1.large.2 2vCPUs 4GB	CentOS 7.6 64bit



华为鲲鹏平台实验环境拓扑





Porting Advisor移植PostgreSQL

1. 检查前提条件是否满足

- ✓ x86云服务器和鲲鹏云服务器已上电
- ✓ x86云服务器、鲲鹏云服务器与PC之间网络互通，且能访问外网
- ✓ 云服务器上已安装CentOS 7.6操作系统，且已获取root用户帐号和密码
- ✓ x86云服务器上已安装Porting Advisor工具
- ✓ 鲲鹏云服务器上已安装5.3及以上版本的GCC编译器
- ✓ PC上已安装SSH远程登录工具putty和WinSCP

2. Porting Advisor代码迁移工具分析源码

- ✓ 上传PostgreSQL源码
- ✓ 创建代码分析任务
- ✓ 根据分析报告和修改建议，修改源码

3. 安装PostgreSQL

- ✓ 生成Makefile文件
- ✓ 编译
- ✓ 安装

4. 启动和连接数据库

- ✓ 创建用户和用户组
- ✓ 初始化数据库
- ✓ 启动数据库
- ✓ 登录数据库
- ✓ 修改配置数据库配置文件
- ✓ 重启数据库服务



BenchmarkSQL安装

1. 检查前提条件是否满足

- ✓ x86云服务器已上电
- ✓ x86云服务器与PC之间网络互通，且能访问外网
- ✓ 云服务器上已安装CentOS 7.6操作系统，且已获取root用户帐号和密码
- ✓ x86云服务器上已安装jdk
- ✓ PC上已安装SSH远程登录工具putty和WinSCP

2. 安装性能测试工具BenchmarkSQL

- ✓ 配置环境变量
- ✓ 获取BenchmarkSQL源码



PostgreSQL性能测试

1. 检查前提条件是否满足

- ✓ x86云服务器已上电
- ✓ x86云服务器与PC之间网络互通，且能访问外网
- ✓ 云服务器上已安装CentOS 7.6操作系统，且已获取root用户帐号和密码
- ✓ x86云服务器上已安装BenchmarkSQL性能测试工具
- ✓ 鲲鹏云服务器上已安装PostgreSQL数据库，并已创建数据库用户postgres和数据库tpcc
- ✓ PC上已安装SSH远程登录工具putty和WinSCP

2. 连接PostgreSQL数据库

- ✓ 配置环境变量
- ✓ 获取BenchmarkSQL源码

3. 使用BenchmarkSQL进行性能测试

- ✓ 初始化数据
- ✓ 压力测试



PostgreSQL性能调优

1. 检查前提条件是否满足

- ✓ 鲲鹏云服务器已上电
- ✓ 鲲鹏云服务器与PC之间网络互通，且能访问外网
- ✓ 云服务器上已安装CentOS 7.6操作系统，且已获取root用户帐号和密码
- ✓ x86云服务器上已安装BenchmarkSQL性能测试工具
- ✓ 鲲鹏云服务器上已安装PostgreSQL数据库，并已创建数据库用户postgres和数据库tpcc
- ✓ PC上已安装SSH远程登录工具putty和WinSCP

2. 使用BenchmarkSQL进行性能调优

- ✓ 编译参数调优
- ✓ 数据库参数调优

3. 使用BenchmarkSQL重新进行性能测试

- ✓ 删除原有的测试数据
- ✓ 初始化数据
- ✓ 压力测试



PostgreSQL软件打包

- 在鲲鹏云服务器上，使用rpmbuild工具制作的PostgreSQL的RPM包，便于后续快速安装PostgreSQL，制作RPM包的步骤如下：





PostgreSQL软件包安装测试

- 在鲲鹏云服务器上，使用制作完成的RPM包安装PostgreSQL，安装RPM包的步骤如下：

获取RPM包

- ✓ 直接将生成的RPM包拷贝至任意目录下

使用命令安装PostgreSQL

- ✓ 使用rpm命令直接安装RPM包



思考题

1. Porting Advisor工具在移植源码过程中的作用是？（ ）
 - A. 分析源码，并给出移植工作量
 - B. 分析源码，并给出分析报告和源码修改建议
 - C. 分析源码，并修改源码
 - D. 分析源码，并给出性能优化建议



本章总结

- 本章主要以PostgreSQL为例，介绍在华为鲲鹏平台上对应用软件进行移植调优的全流程综合实验，包括环境搭建、软件移植、性能测试、性能调优、软件打包等。

The image features a blue-tinted background with silhouettes of several groups of business professionals in a modern office environment. They are engaged in various activities like reviewing documents and talking. The overall aesthetic is professional and corporate.

谢谢

www.huawei.com



鲲鹏解决方案



前言

- 本课程给大家介绍华为鲲鹏解决方案，全面介绍华为鲲鹏解决方案在通用解决方案和行业解决方案的能力。



目标

- 学完本课程后，您将能够掌握：
 - 华为云鲲鹏解决方案包含哪些通用解决方案
 - 华为云鲲鹏解决方案包含哪些行业解决方案



目录

- 1. 华为鲲鹏解决方案全景**
2. 华为鲲鹏通用解决方案
3. 华为鲲鹏行业解决方案



华为鲲鹏解决方案全景

- 基于华为鲲鹏计算平台（云平台或物理平台）的解决方案分为通用解决方案及行业解决方案，通用解决方案包括全栈专属云HCS Online、高性能计算HPC、大数据基础设施、分布式存储、企业核心应用等，行业解决方案包括运营商、政府、金融等。
- 部署方式可以全部采用鲲鹏平台部署，也可以采用鲲鹏平台和X86平台混合部署的方式。

行业解决方案



运营商



政府



金融



游戏



媒体与娱乐



智能安防

通用解决方案



全栈专属云HCS Online



高性能计算HPC



大数据基础设施



鲲鹏分布式存储



企业核心应用

华为鲲鹏计算平台

华为云鲲鹏云服务





华为鲲鹏云全栈解决方案竞争力

行业 ISV



政府 金融 能源 交通 医疗 教育

鲲鹏云解决方案

企业SAAS (JAVA/Python/Tomcat/Mysql) 企业Web (Apache/TomCat/Nginx) 国产应用 (JAVA/Pyhon/东方通/达梦) 传统企业应用 (Tomcat/redis/postgresql)

编译工具: GCC, OpenJDK, PHP, Python; 数据库: Mysql, Postgresql, MariaDB, MongoDB; 中间件: 东方通, 金蝶天燕, Tomcat, 方正飞鸿

华为鲲鹏云服务

其它操作系统 Eluer OS
ECS BMS EVS RDS CCE ELB VPC CCI

生态丰富

300+ 企业应用适配鲲鹏云服务

全栈能力

100+ 常用组件适配鲲鹏云服务

服务多样

ALL 华为云服务适配支持鲲鹏



TaiShan服务器覆盖主流应用场景

大数据



- 性能提升
- 加解密引擎
- 与x86融合部署

分布式存储



- IOPS性能提升
- 压缩解压缩引擎
- 与x86融合部署

数据库



- RoCE低时延网络
- 多核调度算法
- NUMA优化算法

原生应用



- 原生同构，无性能损耗
- 企业级可靠方案
- 百万级应用，成熟生态

HPC



- 八通道内存带宽
- 高密型服务器
- 液冷节能技术

高性能

多核高并发

高吞吐

原生算力同构



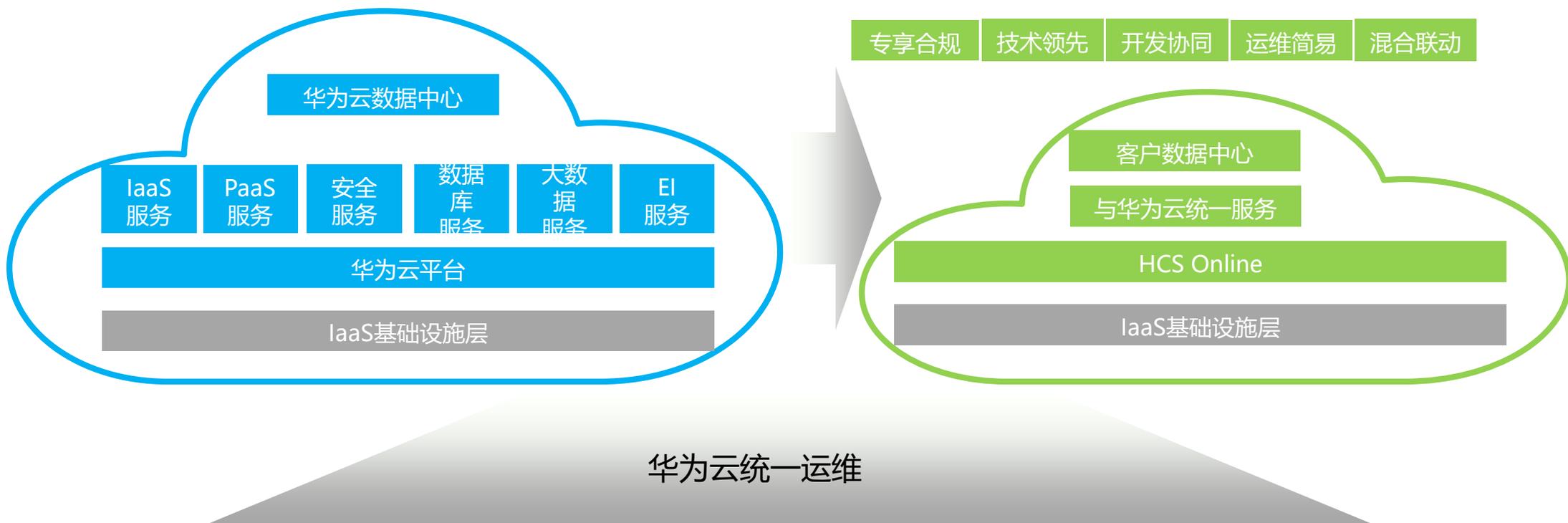
目录

1. 华为鲲鹏解决方案全景
- 2. 华为鲲鹏水平解决方案**
 - 全栈专属云HCS Online解决方案
 - 高性能计算HPC解决方案
 - 大数据基础设施解决方案
3. 华为鲲鹏行业解决方案



HCS Online解决方案概述

- 全栈专属云HCS Online解决方案 (Huawei Cloud Stack Online, HCS Online) 是华为云在客户数据中心的一种延伸，以一体化全栈方式交付完整的云服务平台，与华为云统一架构、统一服务、统一API。
- HCS Online部署在客户数据中心，物理隔离以确保安全合规，管控面通过专线接入华为云运维中心，统一运维以降低运维成本，是政府、金融、以及大企业客户核心业务上云的优选方案。





HCS Online解决方案核心价值

享有公有云服务体验

- HCS Online与华为云统一架构，继承公有云大规模商用成熟架构，稳定可靠、开放兼容。
- HCS Online提供与华为云一致的API和管理工具，降低客户开发适配成本。
- HCS Online服务能力与华为云同步，云服务之间互相解耦，可根据业务场景按需部署、平滑扩展、快速同步华为云能力。
- 提供完整的IaaS专属服务。
- 按需提供数据库、大数据、安全服务、中间件和应用等服务。

本地部署、统一运维

- HCS Online部署在客户数据中心，用户独享计算、存储、网络以及云服务等资源，物理隔离以满足特定性能、业务应用和安全合规的要求。
- HCS Online可通过专线接入华为云运维中心，由华为云专业运维团队统一运维，简化运维，降低运维成本，是政府、金融、以及大企业客户核心业务上云的优选途径。

构建同构混合云

- HCS Online可通过VPN或云专线接入华为云，从而构建与华为云统一架构的混合云。可实现统一认证、一键SSO（Single Sign On, 单点登录）、不同云服务平台之间的跳转、跨云容灾备份，业务分层部署（例如分层存储架构）等场景。



HCS Online解决方案产品优势

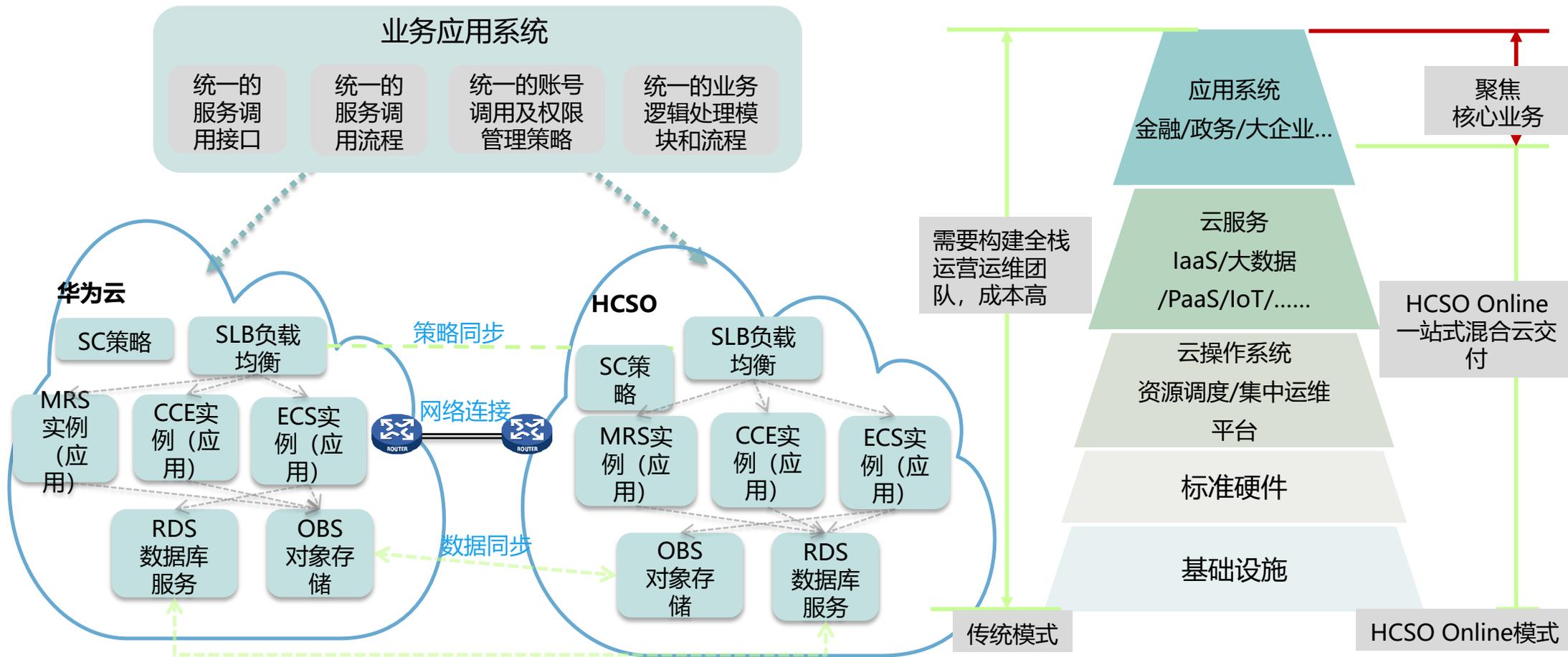
- 聚焦业务应用，与公有云服务长期同步演进
- 一体化硬件、全栈云服务软件、统一维护升级的整体架构
- 面向业务场景持续丰富的云服务能力
- 部署模式灵活，满足不同场景需求
- 全栈防护体系
- 完整灾备能力，全面提升业务连续性
- 专属迁移与运维服务



聚焦业务应用，与公有云服务长期同步演进

一致的业务开发模式

更加敏捷快速的业务迭代





一体化硬件、全栈云服务软件、统一维护升级

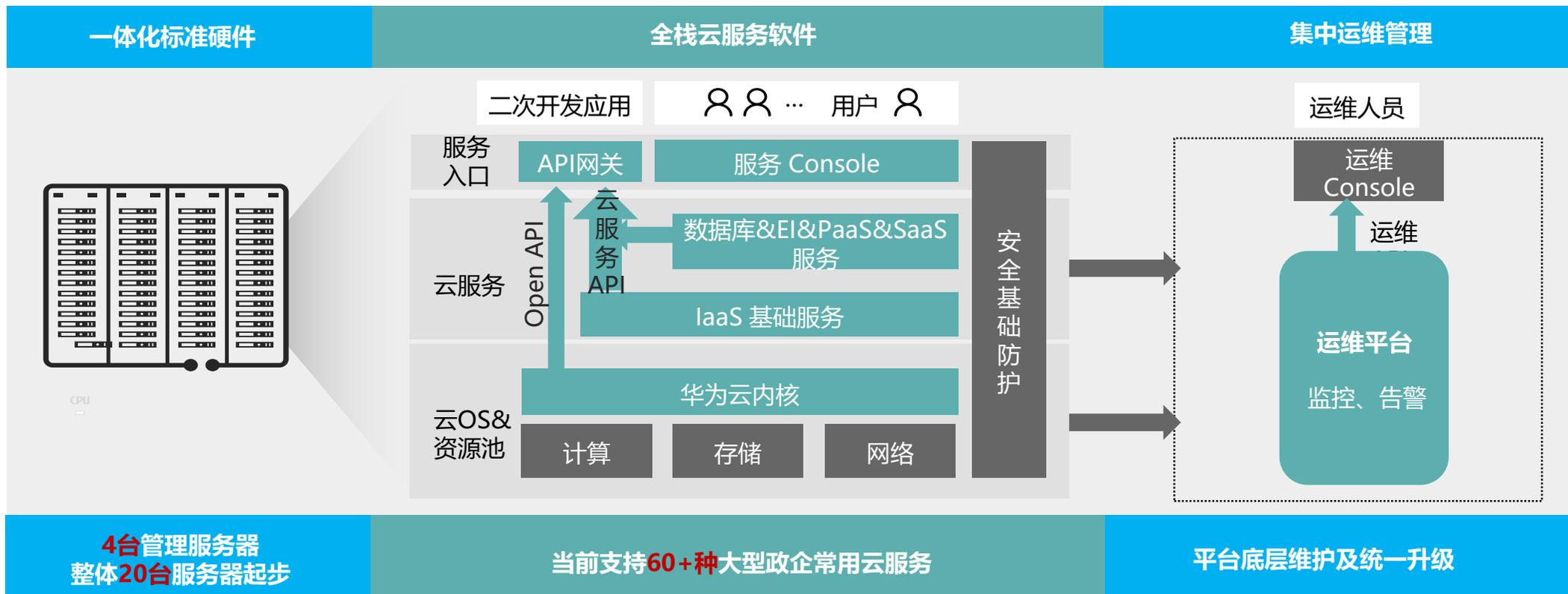
统一架构

统一API

统一云服务

统一生态

统一运维





面向业务场景持续丰富的云服务能力

10+
解决方案

高性能计算

生物医药

云上办公

上云迁移

容灾备份

政府行业

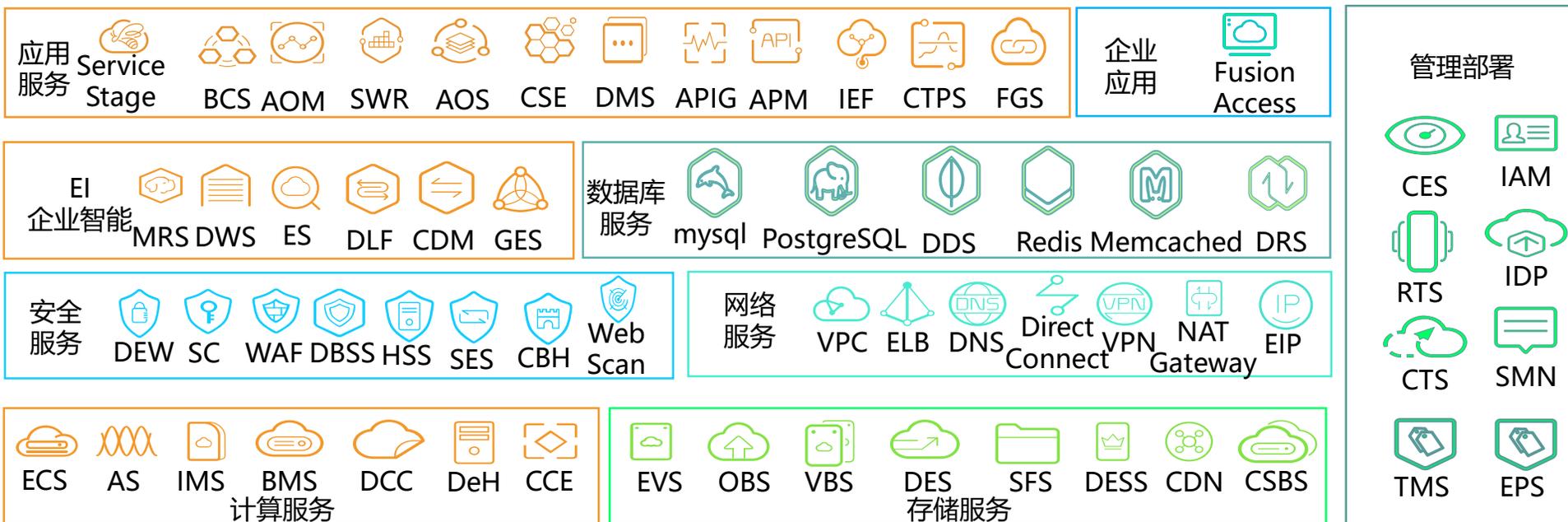
金融行业

医疗影像

零售行业

运营商

60+
服务



42服务

45服务

60+服务

70+服务

2018.8

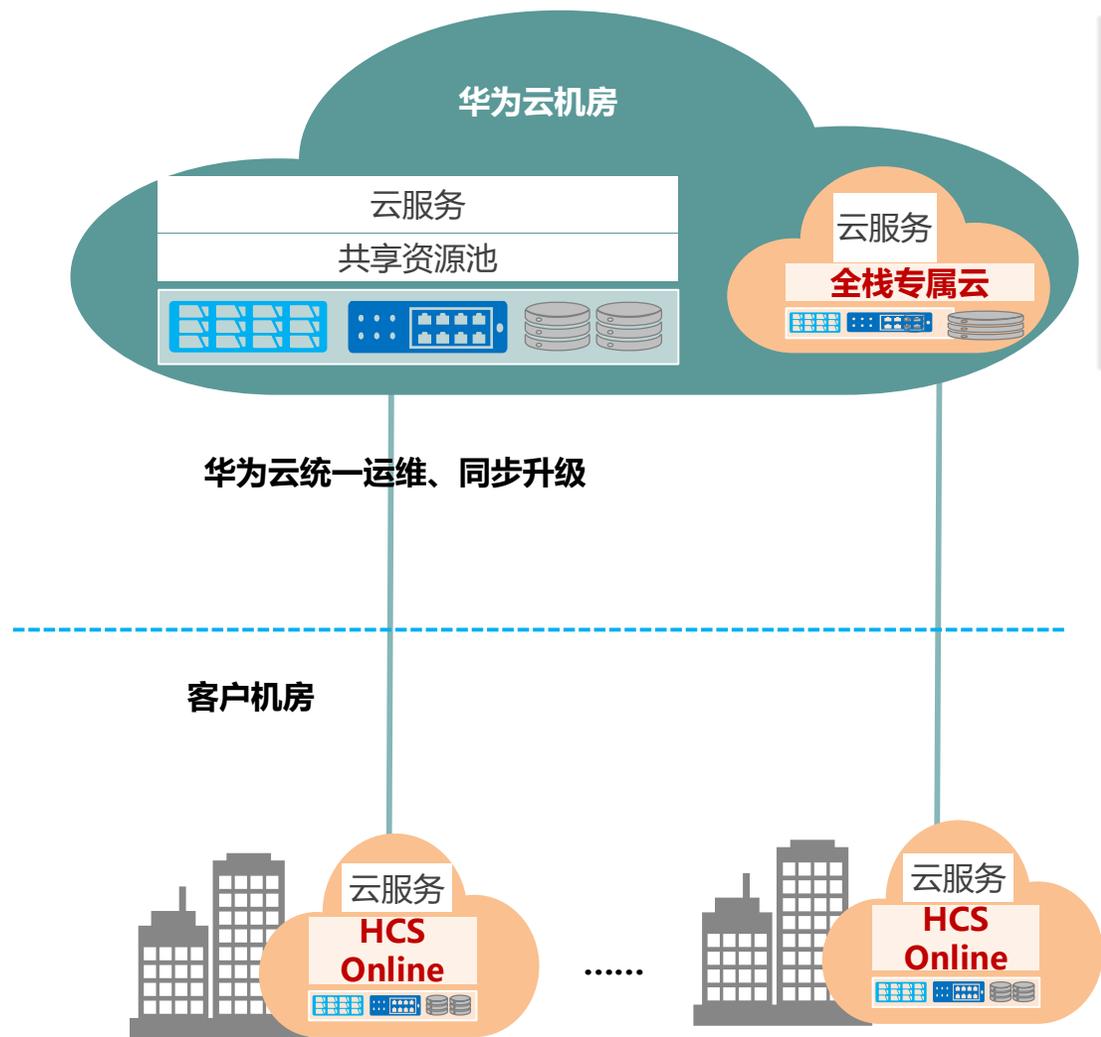
2019.01

2019.03

2019.09



部署模式灵活，满足不同场景需求



全栈专属云

1. 部署位置：**公有云机房**
2. 业务场景：**云服务专属，资源隔离，安全合规，金融专区**
3. 商务模式：**服务租用，云服务按交付容量付费**

HCS Online

1. 部署位置：**客户机房**
2. 业务场景：**数据本地保存，资产安全合规**
3. 商务模式：**硬件买断+云服务按交付容量付费**



全栈防护体系

运营安全

运营管理



运营牌照



租户生命周期



服务生命周期

交易管理



责任边界



合同管理

租户安全

安全服务



自研服务



合作生态

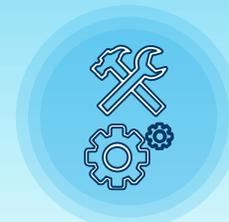
云服务安全

SaaS

PaaS

IaaS

运维安全



检测

分析

感知

响应

基础设施安全

云平台安全



数据安全



主机安全



虚拟网络安全

边界防御安全



防火墙



VPN



WAF



IPS



.....

合规安全 (标准和认证)

安全等保四级

CSA STAR

PCI-DSS

ISO27001

ISO27018

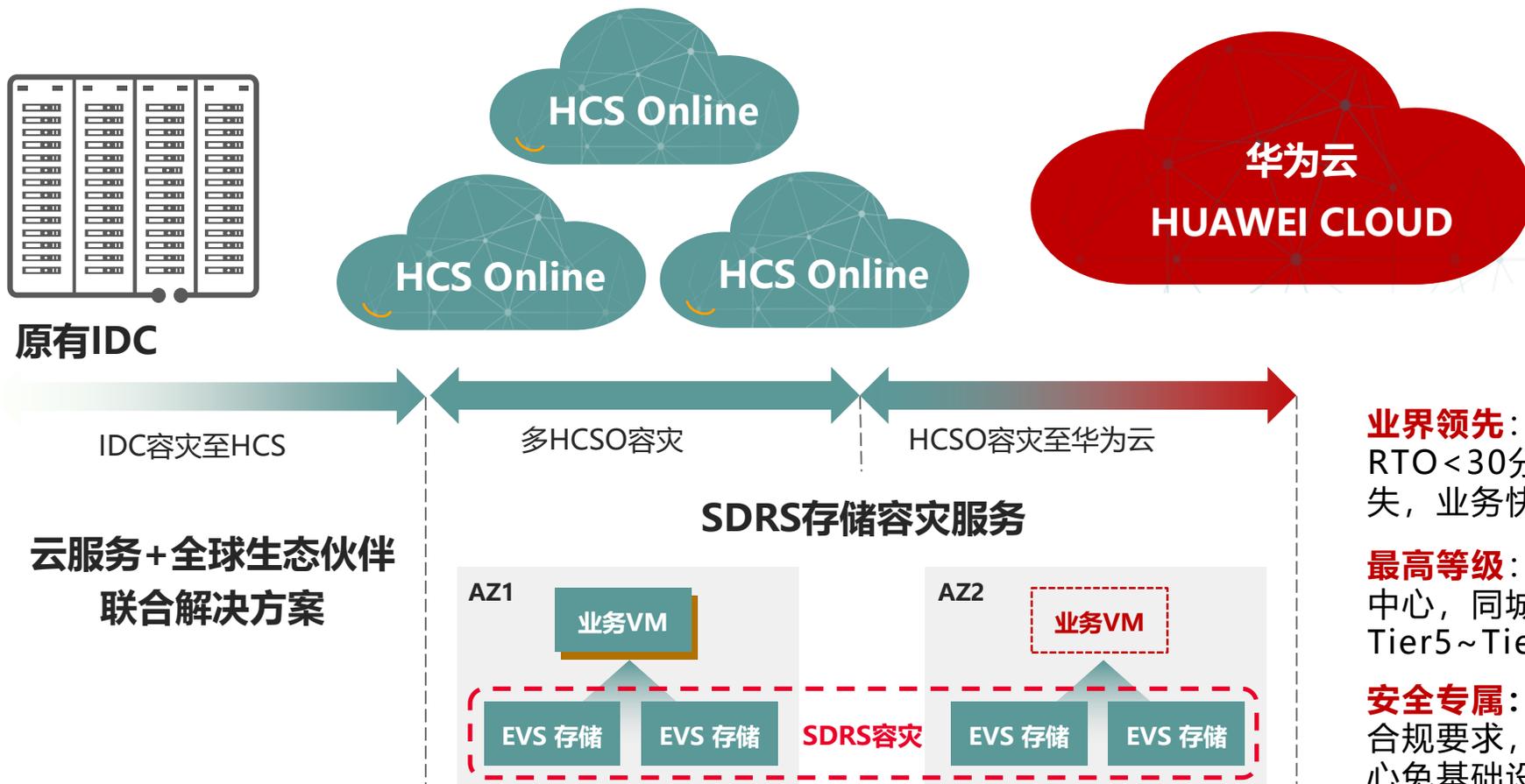
可信云认证

SOC

.....



完整灾备能力，全面提升业务连续性



业界领先：跨AZ RPO=0，RTO<30分钟，确保数据零丢失，业务快速恢复

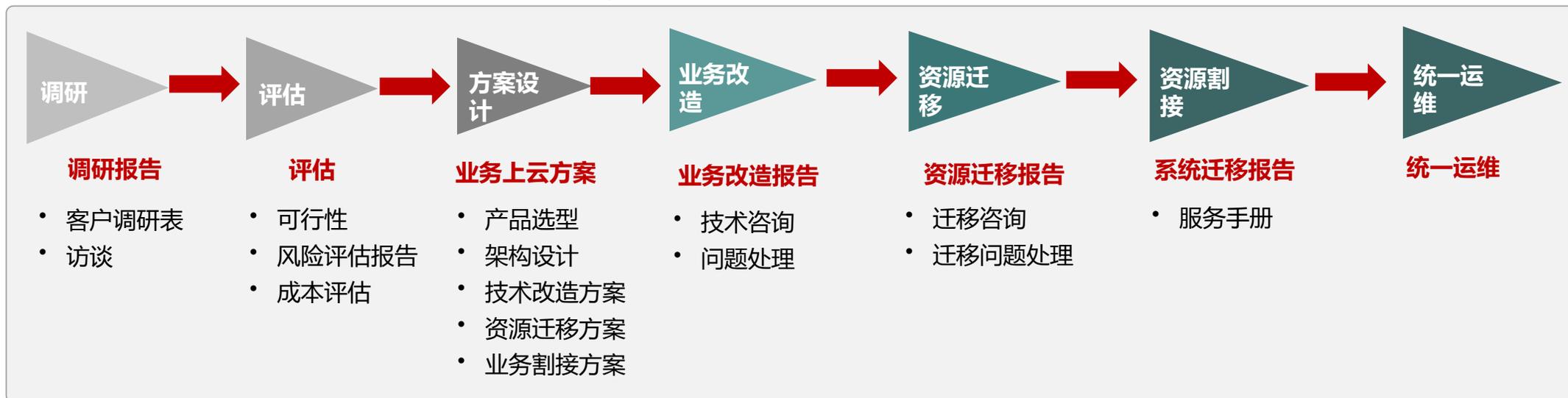
最高等级：快速构建云上两地三中心，同城双中心可达Tier5~Tier6容灾等级

安全专属：物理隔离，满足安全合规要求，相比客户自建容灾中心免基础设施运维



专属迁移与运维服务

- 完善的迁移流程和方法论为用户提供专业迁移服务。
- 提供华为云产品整合迁移能力和云化转型咨询能力。
- 专业服务团队与最佳实践能力为客户提供99.95不中断服务。





HCS Online 主要应用场景

行业云场景

开放接口
服务对接

本地门户
按需定制

分权分域
精细管理

内部结算清晰透明，
定制化提升客户满意度

HCSO+HPC场景

最强HPC
计算性能

联合ISV
构建生态

灵活调度
简易部署

最佳HPC性能，改善TTM，降低
人力成本

HCSO+灾备场景

业界顶尖
存储容灾

华为多年
容灾经验

领先容灾
生态厂商

基于华为云快速建设灾备中心，专
业容灾咨询，复制华为成功经验

智能大数据场景

云上训练
云下推理

最强异构
计算能力

ROMA
数据互通

业务属地化部署，快速使能行业
创新，统一运营运维

多级混合云场景

云平台
统一认证

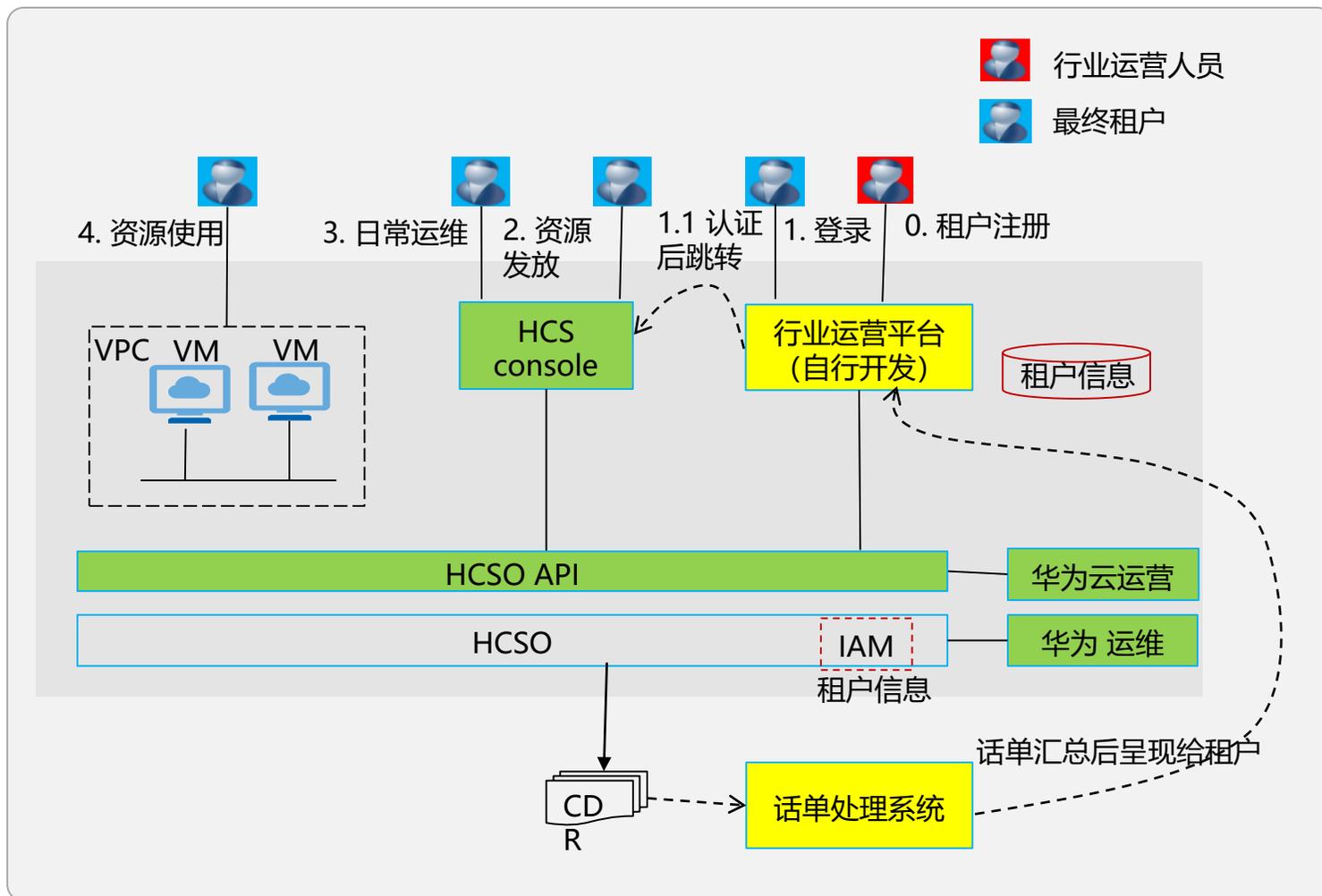
多区域
统一管理

混合云
统一安全

云平台统一建设，体验一致，
灵活按需扩展分级云平台



行业云场景：快速构建行业云，提供先进行业云服务



业务场景

- 场景1: 面向政府的内部结算场景，如政务云平台；
- 场景2: 面向金融保险、医疗、教育、园区；
- 场景3: 运营商的行业云转售场景。

方案优势

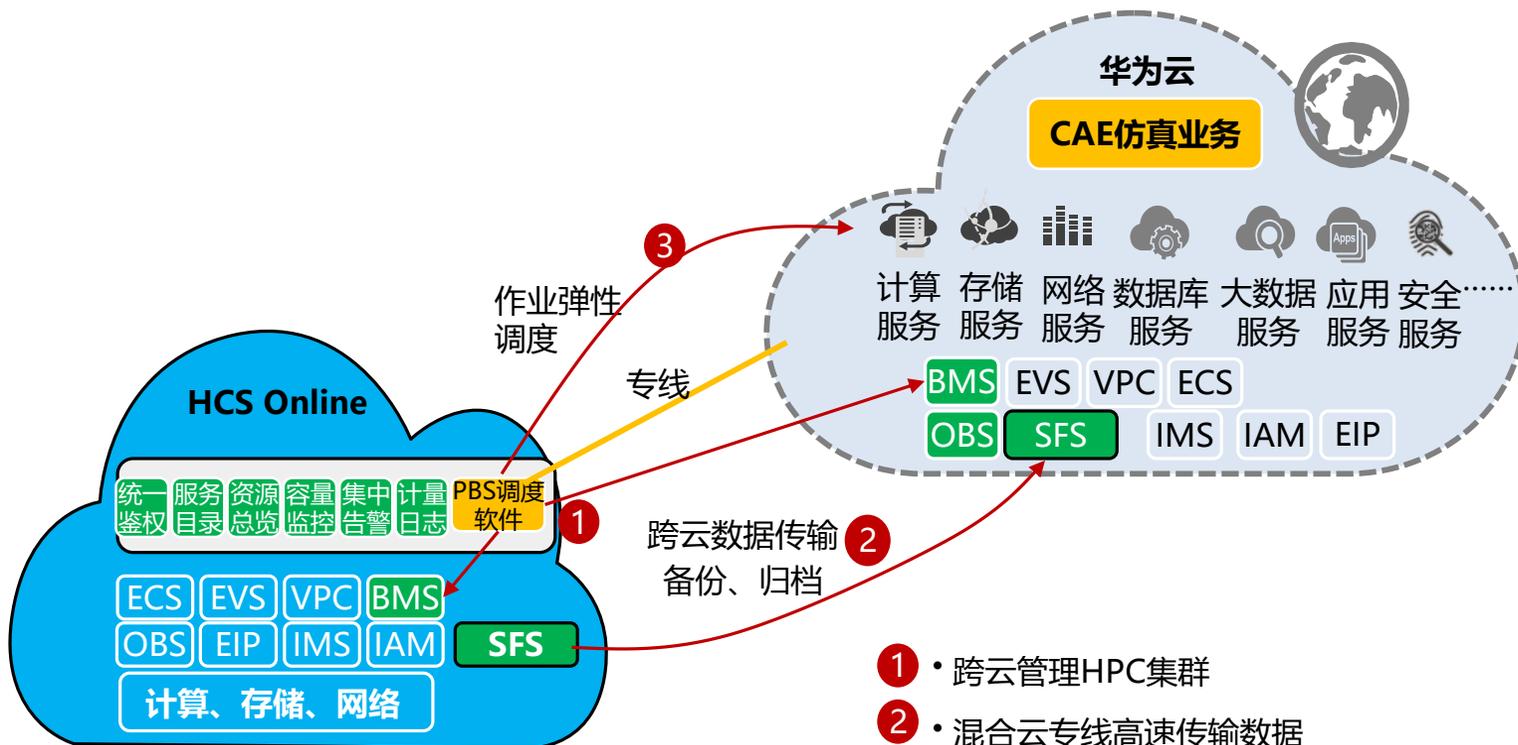
- 政府内部结算清晰透明；
- 为行业客户提供定制服务，提升行业云运营方业内竞争力和最终客户粘性；
- 提升运营商竞争力。

客户价值

- 集中建设，共建共享，统一运营运维，减少投资；
- 丰富的云服务能力和生态；
- 满足智慧园区场景下业务属地化部署。
- 企业数字化、智能化转型，快速使能行业创新。



HCSO+HPC场景:提升HPC性能, 改善TTM



- ① • 跨云管理HPC集群
- ② • 混合云专线高速传输数据
- ③ • 在本地和公有云之间动态调整作业比例

业务场景

- 场景1: 大企业高性能计算场景;
- 场景2: 高校、研究所科研超算场景;
- 场景3: 智能超算场景;

方案优势

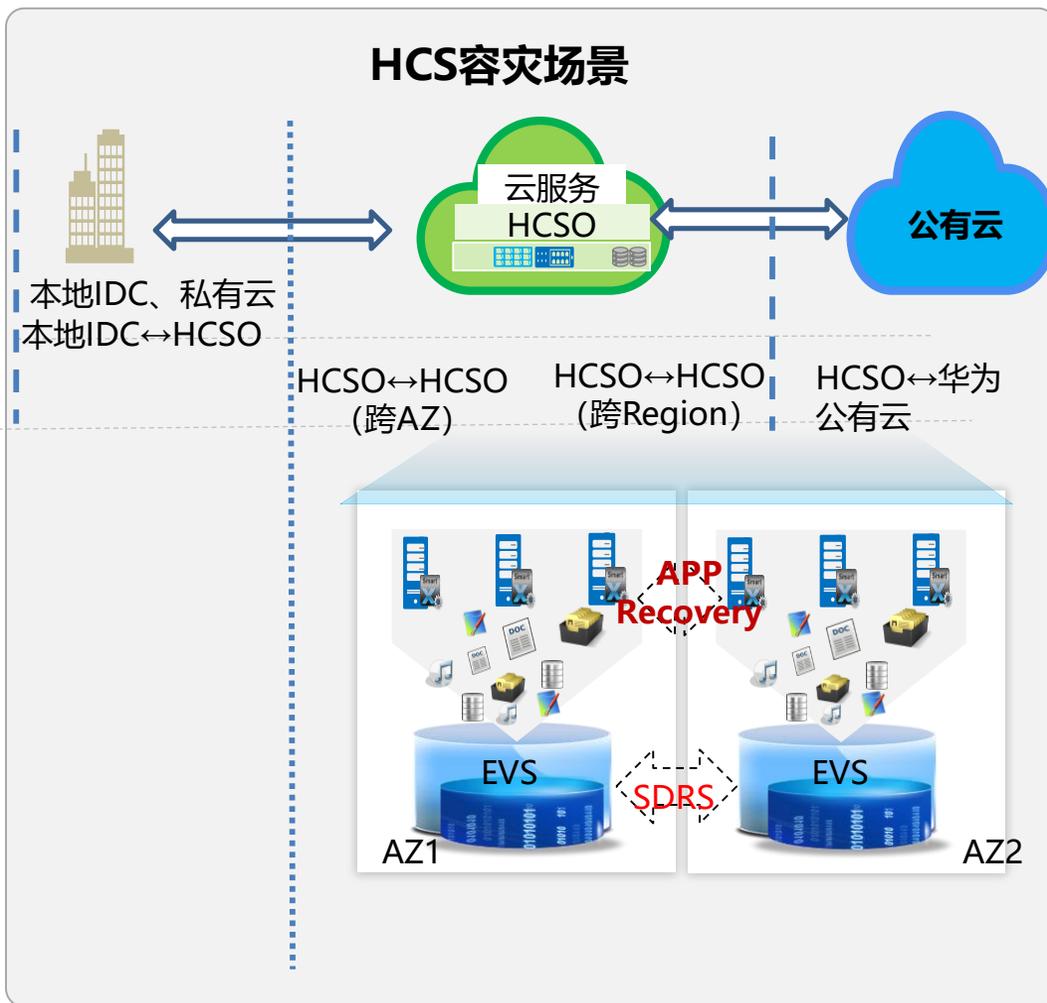
- 最强HPC计算性能;
- 全堆栈HPC解决方案, 联合Altair、并行科技、联科等ISV构建HPC混合云、基于业务流的全场景解决方案;
- HPC任务批处理调度能力;
- HPC业务集群一键部署能力;

客户价值

- 最佳HPC性能, 改善TTM;
- 业务便捷开展, 降低人力投入成本;



灾备场景：跨域灾备，两地三中心构建差异化优势



业务场景

- **场景1:** 有业务连续性和安全合规性高要求的政府部门打造专属云容灾中心。
- **场景2:** 面向金融保险、医疗、教育、园区，保障业务的持续高可用。

方案优势

- **业界领先:** 跨AZ RPO=0, RTO < 30分钟，确保数据零丢失，业务快速恢复。
- **最高等级:** 快速构建云上两地三中心，同城双中心可达Tier5~Tier6容灾等级。
- **专属安全:** 物理隔离，满足安全合规要求，相比客户自建容灾中心免基础设施运维。

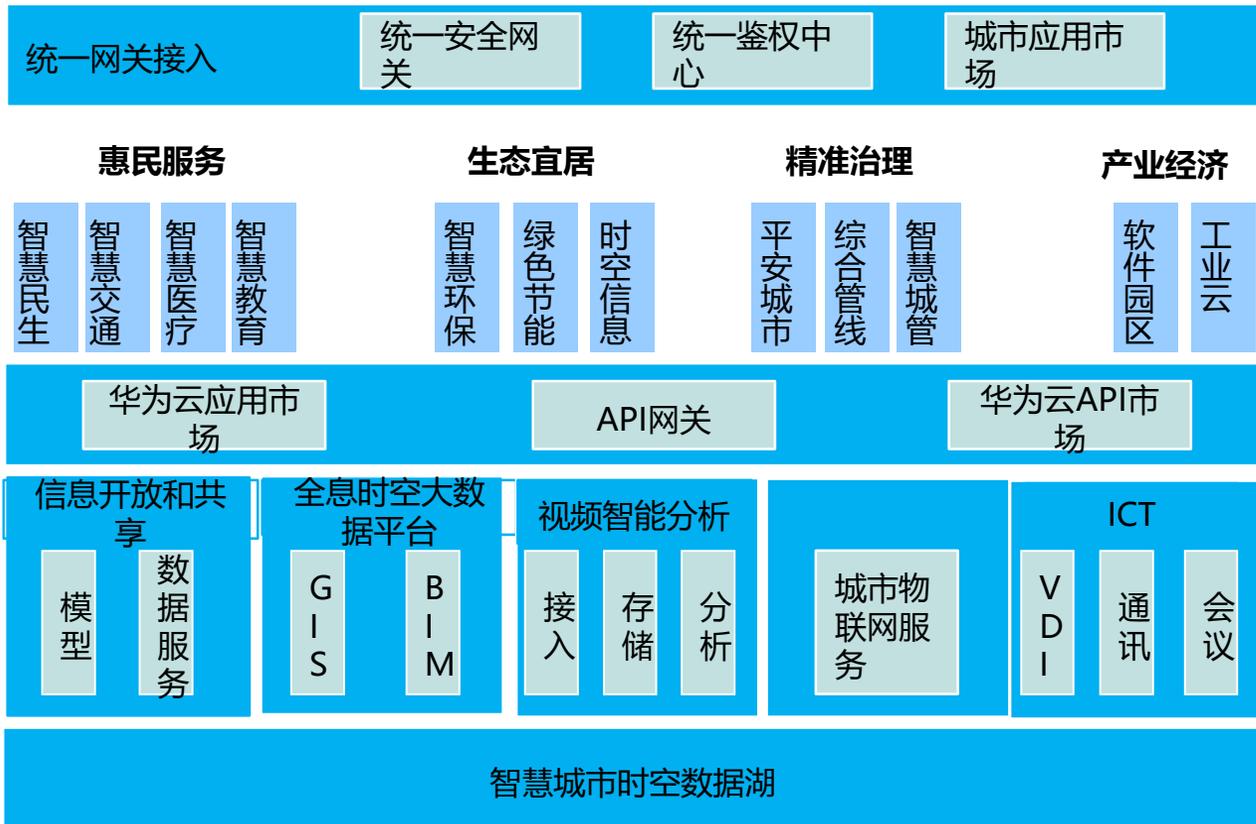
客户价值

- 基于华为云多地域数据中心快速建设跨地域灾备中心，在超大规模地域级自然灾害的时候都能保护数据和业务，满足业务连续性要求。
- 用户独享物理资源，全栈物理隔离，满足客户安全合规要求。
- 专业容灾咨询服务，复制华为高业务连续性成功经验。



智能大数据场景：云上云下智能混合

智慧城市市场



机器学习服务
MLS



深度学习
DLS



MapReduce服务
MRS



数据仓库服务
DWS



数据可视化
DLF



联接服务
ROMA



智能边缘平台
IEF

业务场景

- 场景1: 面向智慧城市市场, 提供城市惠民服务、生态宜居服务、精准城市治理服务、产业与经济服务等;
- 场景2: 企业大数据应用平台

方案优势

- EI分级部署, 云上训练, 云下推理
- 华为云语音识别、人脸识别、OCR的开放API能力
- ROMA解决方案集成能力
- IEF最强边缘计算能力

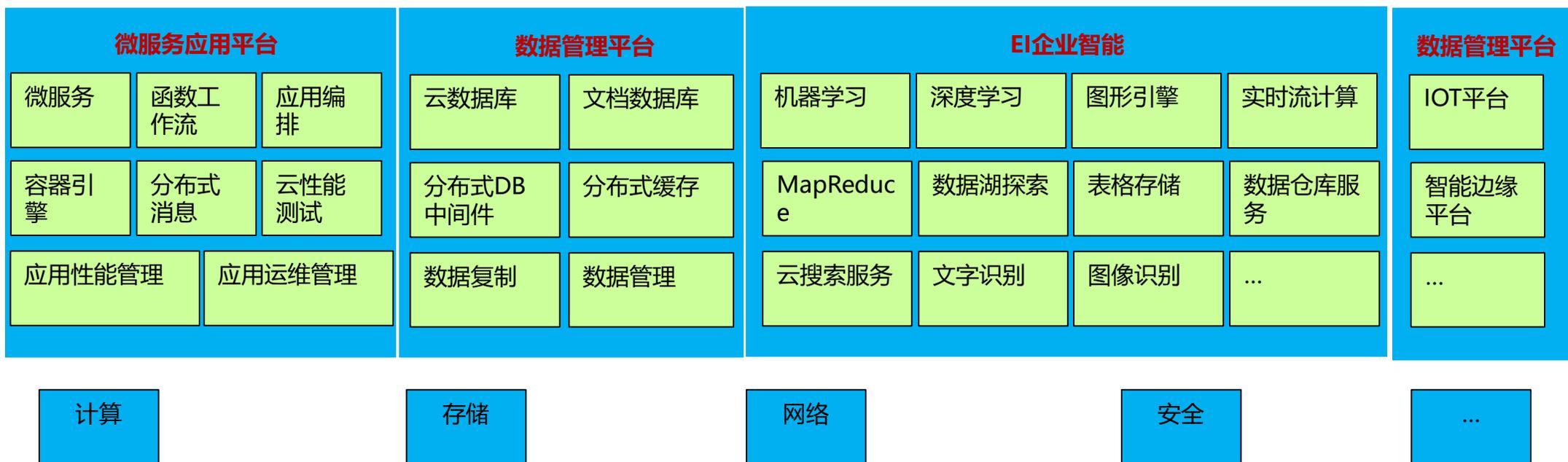
客户价值

- 集中建设, 共建共享, 统一运营运维
- 丰富的云服务能力和生态
- 满足智慧城市市场下业务属地化部署
- 企业数字化、智能化转型, 快速使能行业创新



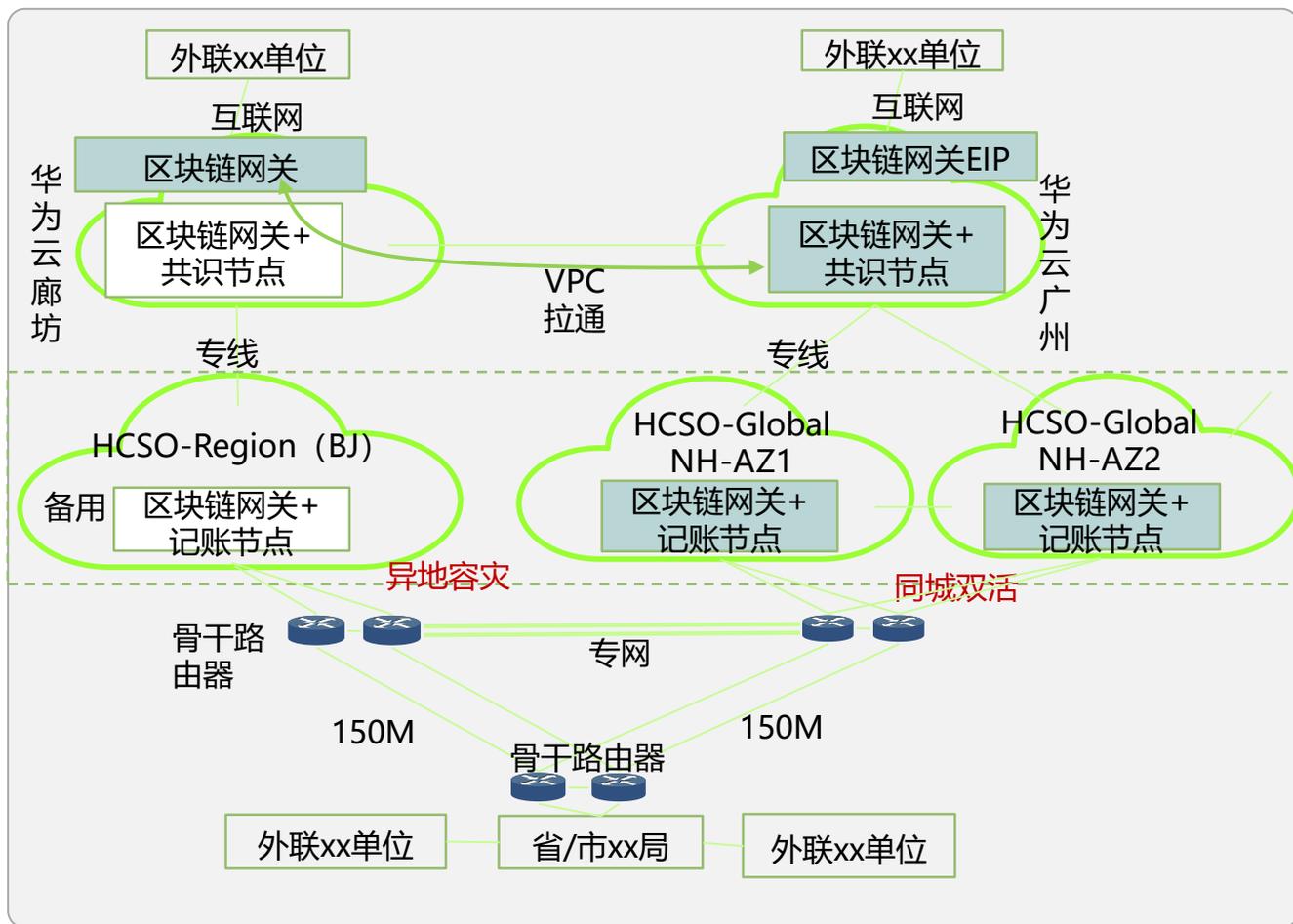
智能大数据场景：云上云下智能混合

华为云HCSO平台





多级混合云场景：统一门户、统一管理、统一安全



业务场景

- 场景1：面向垂直部委安全合规场景，如财税、海关等
- 场景2：面向政府公共安全场景，如雪亮工程、智慧交警、检法司安等
- 场景3：企业集团云平台统一建设，构建集团-分支机构多级扩展云平台

方案优势

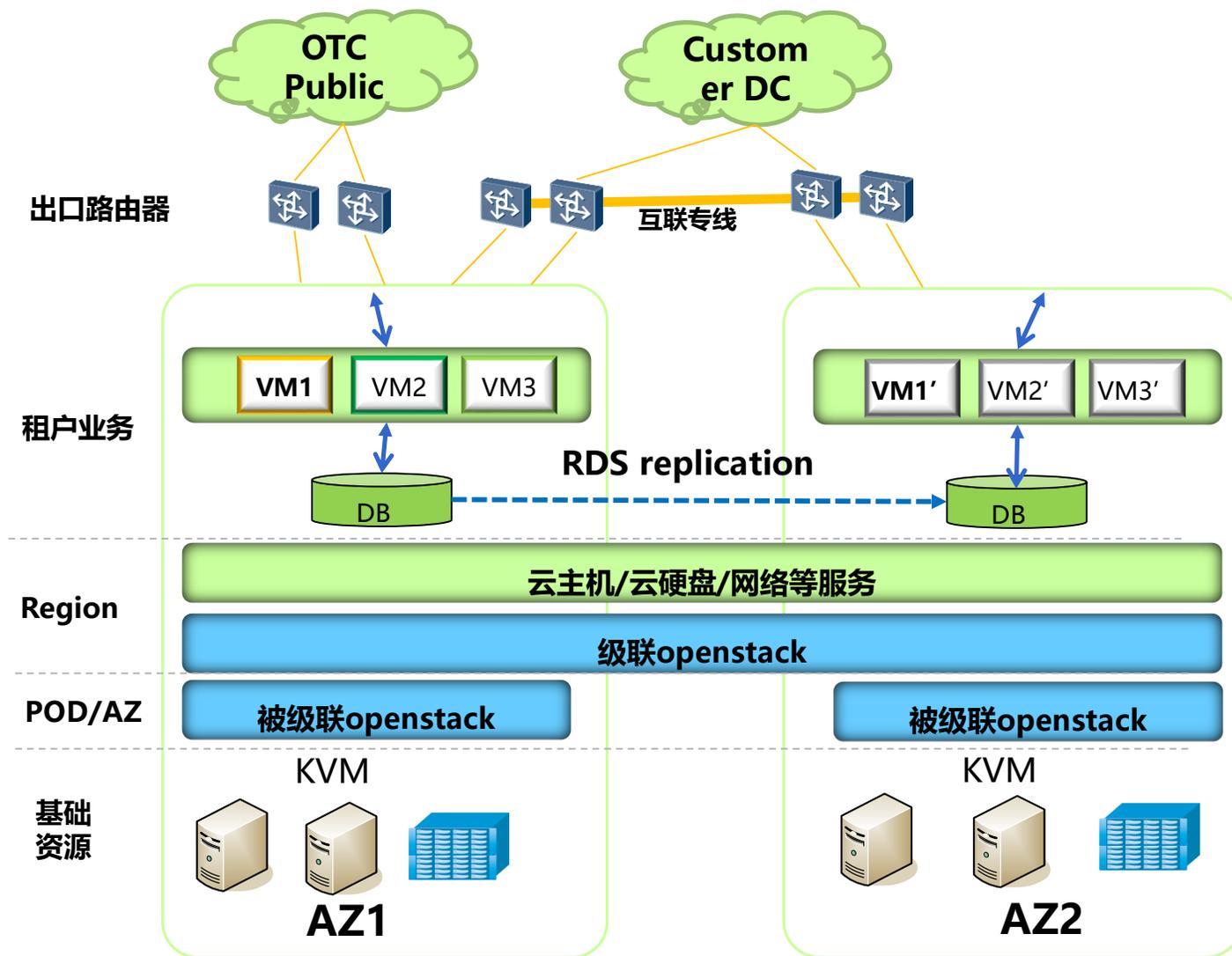
- HCS Global+Region部署构建多级同构混合云
- 联邦认证实现云平台统一认证，一键SSO
- 智能的混合云DDoS协同防护能力，业务分层部署
- 软硬一体的全栈纵深防护安全体系
- 统一门户、统一运维、统一安全

客户价值

- 满足政府机构业务和数据安全合规要求
- 协助政府IT系统安全架构顶层设计和规划
- 云平台统一建设，使用体验一致
- 灵活按需扩展分级云平台



案例：欧洲某铁路集团双AZ混合云



解决方案

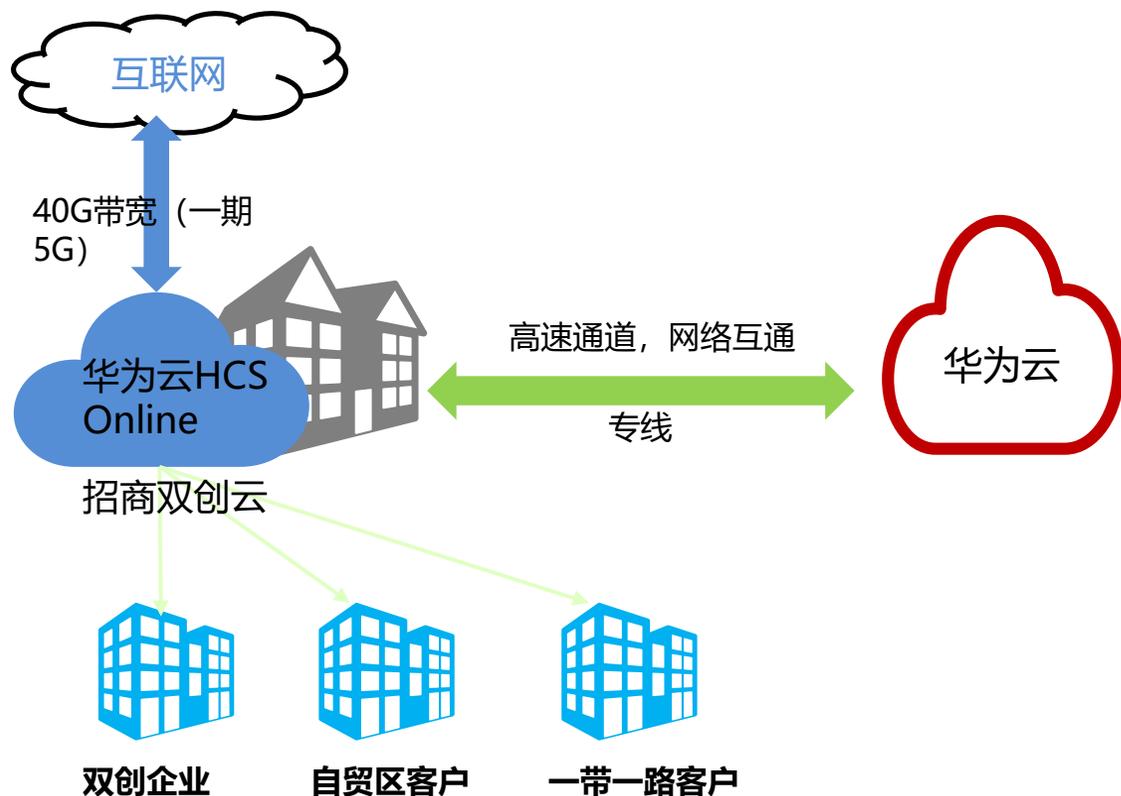
- 双AZ备份容灾
- HCS Online和OTC公有云统一运维运营
- HCS Online专享独占，安全合规
- 云服务模式方便快捷

客户价值

- 高性价比、轻量化起步，省事省心
- 按需采购，快速响应
- 安全合规，专属隔离
- 运维运营，弹性扩展



案例：招商双创园区企业创业创新



解决方案

- 基于HCS Online方案与天翼云结合，为招商打造一朵招商双创云；
- 双创云接入天翼云的二级运营运维，集约化部署；
- 企业专线接入；

客户价值

- 企业拎包入住；
- 依托电信及天翼云，合规运营；
- 云网融合：云专线提供一站式打包服务；



目录

1. 华为鲲鹏解决方案全景
- 2. 华为鲲鹏水平解决方案**
 - 全栈专属云HCS Online解决方案
 - 高性能计算HPC解决方案
 - 大数据基础设施解决方案
3. 华为鲲鹏行业解决方案



高性能计算HPC解决方案概述

- 高性能计算HPC (High Performance Computing, 高性能计算) 是一个计算机集群系统, 它通过各种互联技术将多个计算机系统连接在一起, 利用所有被连接系统的综合计算能力来处理大型计算问题, 所以又通常被称为高性能计算集群。
- 高性能计算云解决方案 (HPC Cloud) 是利用云部署方式来提供一种高效、可靠、灵活、安全的计算服务; 能够为工业设计、海量数据处理等场景提供卓越的计算服务; 帮助客户降低TCO, 缩短产品上市周期, 提升企业产品竞争力。
- 华为HPC解决方案主要分两层:
 - 华为主要提供相关的计算所需要的IAAS资源, 包括计算、存储、网络等资源。
 - 集群调度层在不同的应用场景使用的软件略有不同, 如工业仿真场景更多的使用的PBS, 而针对基因测序使用的更多的是开源软件Sge以及Slurm。



高性能计算HPC解决方案核心价值

- 部署周期短，成本低。
- 部署灵活，不会造成资源的浪费或不足。
- 公有云方式维护成本低。
- 满足技术快速发展要求。
- ARM HPC瞄准内存敏感型应用及整型计算（或者单精度计算）的应用，比如气象环保，基因测序。



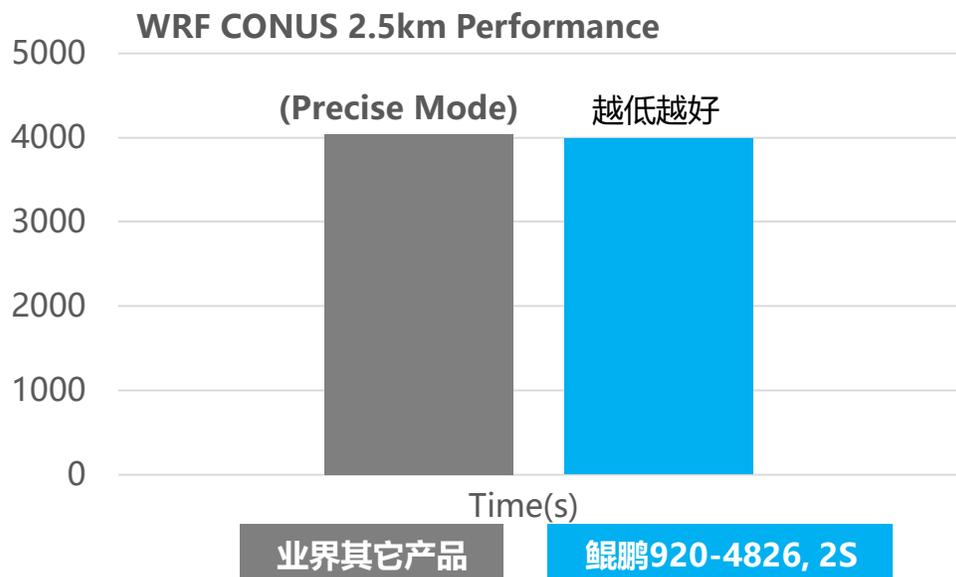
高性能计算HPC解决方案优势

- 性能卓越
- 性价比高：方便易用、生态丰富
- 多组织、区域租户共享
- 自服务能力
- 部署方式灵活

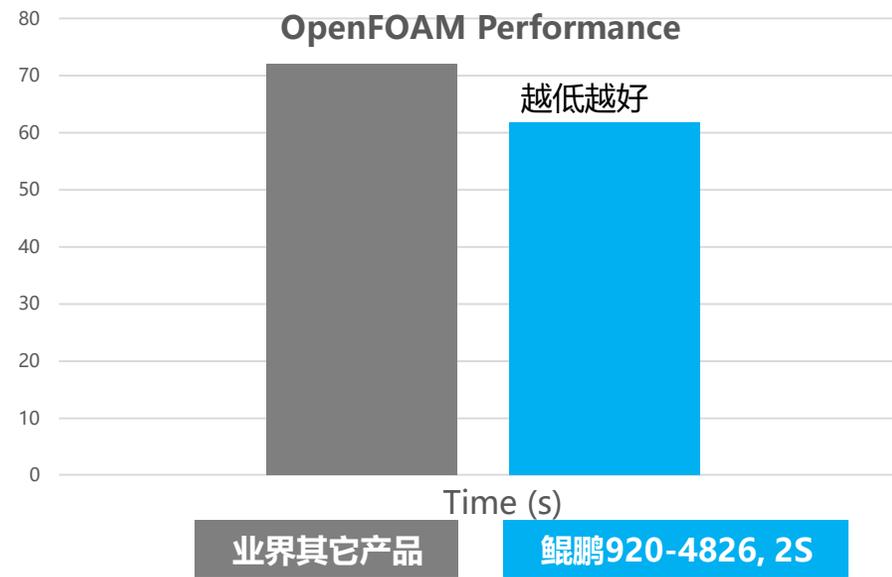


性能卓越

气象类应用



CFD类应用



	业界其它产品	鲲鹏920-4826, 2S
OS	CentOS 7.5	openEuler
MPI	Intel MPI 18.0.1	Open MPI 3.0.0
Compiler	icc 18.0.1	ahc18.4
Time (s)	4039	3989

	业界其它产品	鲲鹏920-4826, 2S
OS	CentOS 7.5	openEuler
MPI	Intel MPI 18.0.1	Open MPI 3.0.0
Compiler	icc 18.0.1	ahc18.4
Time (s)	72.03	61.74

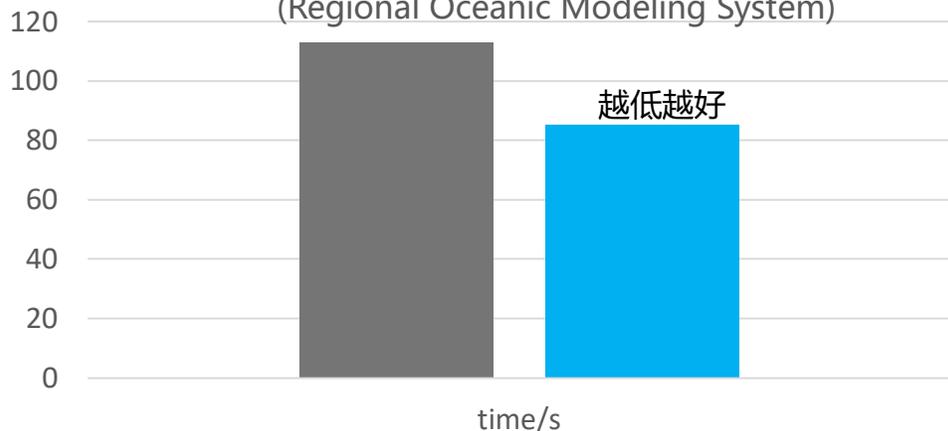


性能卓越

气象类应用

ROMS Performance

(Regional Oceanic Modeling System)



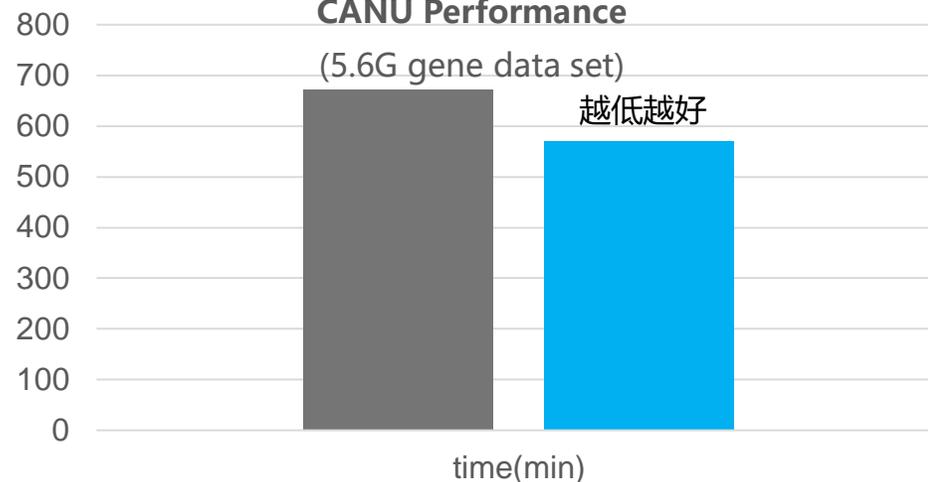
业界其它产品

鲲鹏920-4826, 2S

基因类应用

CANU Performance

(5.6G gene data set)



业界其它产品

鲲鹏920-4826, 2S

	业界其它产品	鲲鹏920-4826, 2S
OS	CentOS 7.4	openEuler
MPI	Intel MPI 18.0.1	Open MPI 3.0.0
Compiler	icc 18.0.1	ahc18.4
Time (s)	201	198

	业界其它产品	鲲鹏920-4826, 2S
OS	CentOS 7.4	openEuler
MPI	Intel MPI 18.0.1	Open MPI 3.0.0
Compiler	gcc 7	gcc 7
Time (s)	672	570



性价比高

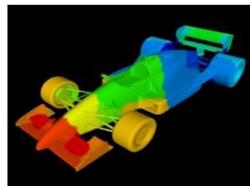
基因测序、流体力学、气象环保主要为**内存敏感型**应用以及**整型计算**（或者单精度计算）为主，适合鲲鹏云HPC应用场景。



基因测序



气象环保



流体力学



科学计算



性价比领先

HPC集群管理控制台

HPC一键式部署

HPC集群管理

HPC作业管理

HPC函数库

HPC编译器、MPI通信库



易部署、易使用、易维护

鲲鹏虚拟机实例

100G IB高速网络

鲲鹏裸金属实例



EVS弹性块存储



OBS对象存储



SFS文件存储



丰富的HPC生态



其它优势特性

多组织、区域 租户共享

- 动态申请，动态共享，计算节点按需申请/释放
- 按需租用，避免过度投资，避免重复建设
- 用户隔离措施保障安全性
- 数据集中，跨组织和区域的合作共享

自服务能力

- 自动发放虚拟机、云化裸机，自动创建集群，长时间自动状态检测
- 将各种不同的HPC应用模板进行初始化导入，在VM模板中部署MPI库、编译库及优化配置等

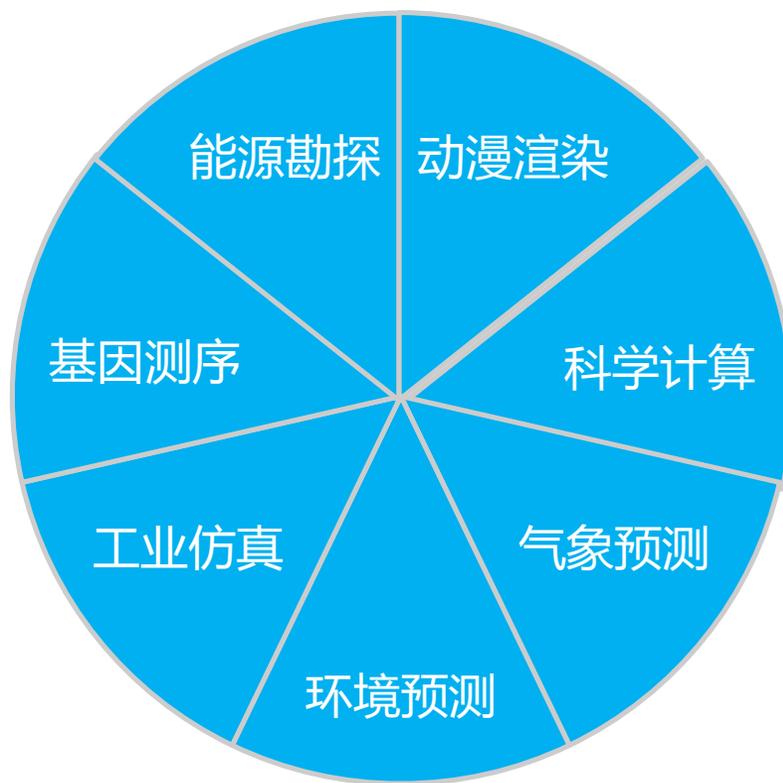
部署方式灵活

- 节省建设周期，近乎无限的基础架构
- 虚机/云化裸机、各计算/存储实例灵活可选



高性能计算HPC解决方案典型业务场景

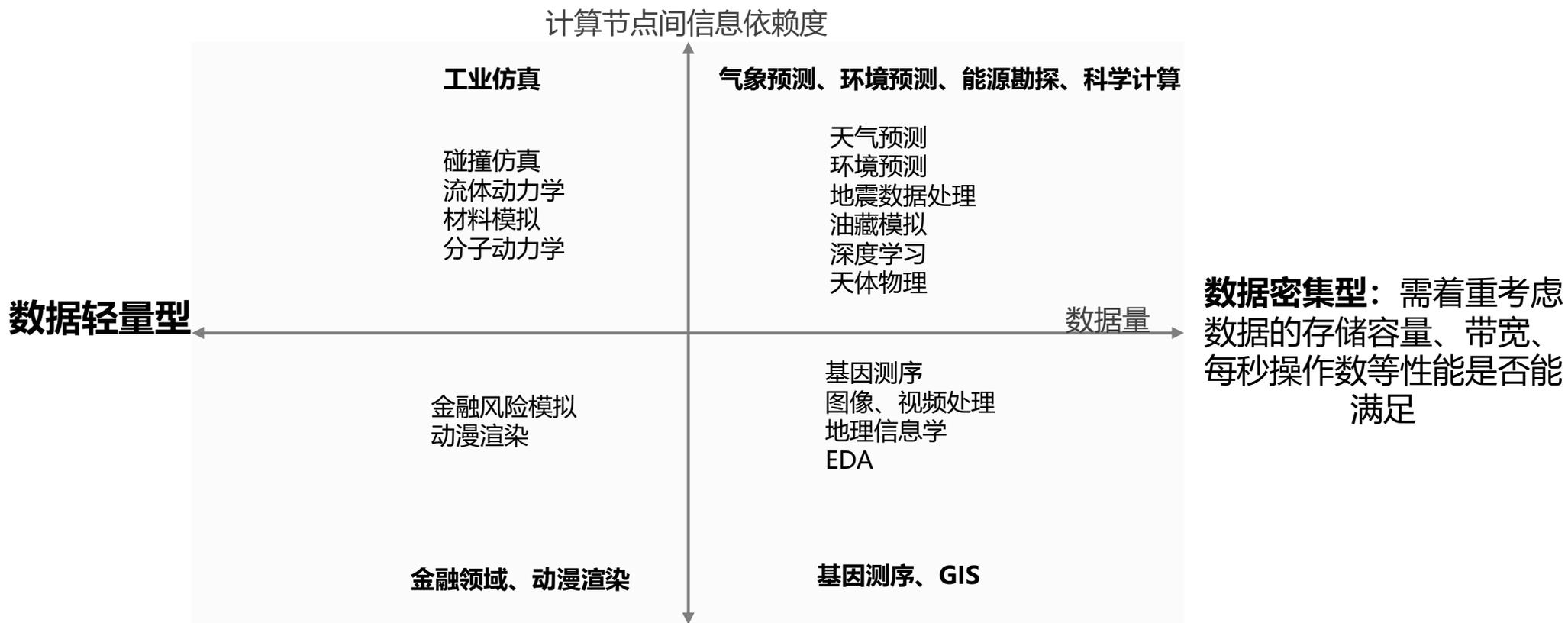
高性能计算HPC解决方案典型场景：科学计算、气象预测、环境预测、工业仿真、基因测序、能源勘探、动漫渲染、金融分析等等。





典型业务场景特点

紧耦合：需配置高带宽低时延的网络



松耦合：采用通用万兆以太网即可支持



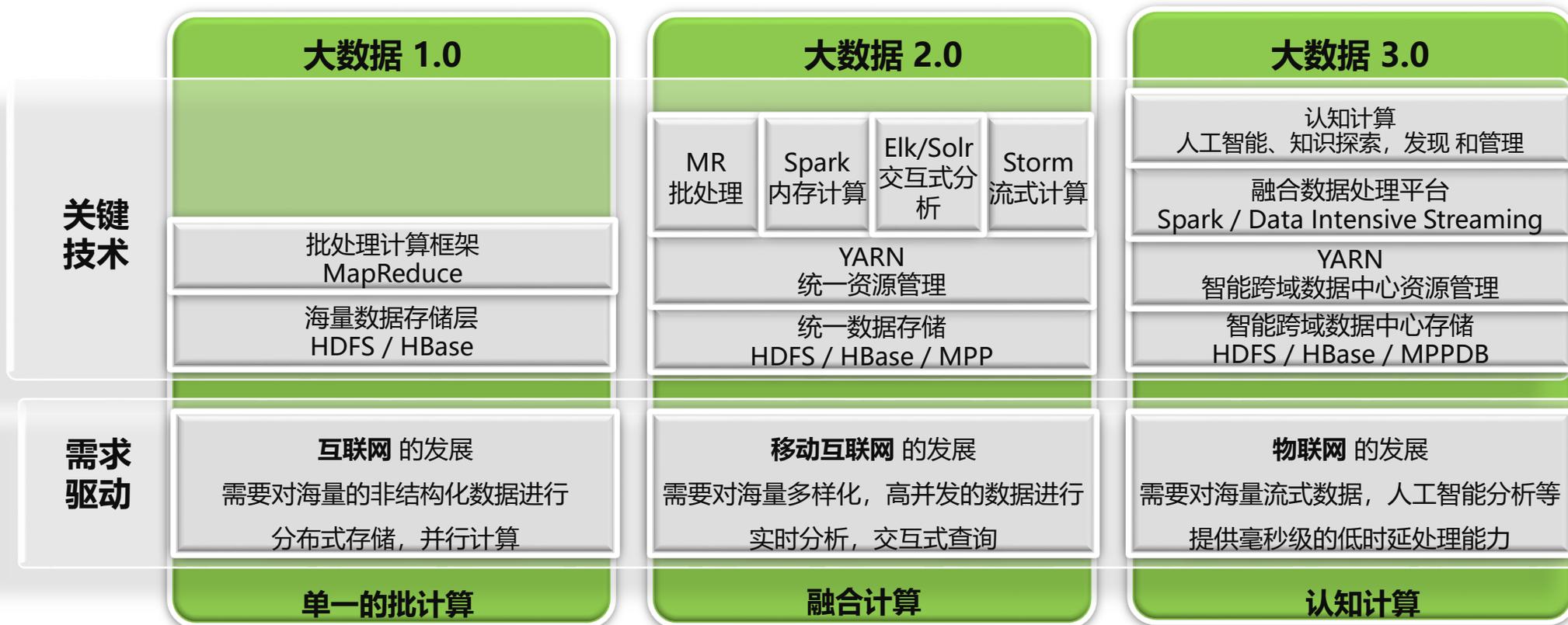
目录

1. 华为鲲鹏解决方案全景
- 2. 华为鲲鹏水平解决方案**
 - 全栈专属云HCS Online解决方案
 - 高性能计算HPC解决方案
 - 大数据基础设施解决方案
3. 华为鲲鹏行业解决方案



大数据基础设施解决方案概述

鲲鹏大数据基础设施解决方案 结合华为云丰富的大数据平台和基础云服务经验，为企业提供高性能、高可靠的大数据业务基础资源、AI训练推理平台，快速实现企业数据化、智能化转型。



海量数据超过单机处理能力

分布式并行计算框架成为标准

高并发度成为加速大数据性能的关键



大数据基础设施解决方案核心价值

高性能

- 鲲鹏(2*920)对比X86双路机型：计算业务性能提升；

芯片级加密

- 鲲鹏920支持国密算法加速，联合FI构建国产化安全可信大数据；
- CPU内置硬件加速器，更安全；
- 加密对业务性能的损耗低于10%；

开放生态

- 支持大数据组件X86与鲲鹏裸金属混合部署方案；
- 支持开源、苏研、星环等多个第三方大数据平台；



大数据基础设施解决方案优势



多元算力

依托鲲鹏多核能力，性价比提升；原生支持X86与ARM芯片。

存算分离，灵活弹性

支持计算、存储分离架构，最大化提升资源利用率，按需使用计算资源；极速扩容伸缩，满足业务量高速增长需求。

极致弹性，智能从容

Serverless架构，算力按需获取，容器秒级发放，业务秒级伸缩，最大化资源利用率；原生支持Kubernetes和Docker生态。

极致性能，稳定可靠

海量分布式云存储OBS 带来超大带宽、更低时延，多计算集群可分析使用同一份数据，效率提升。



目录

1. 华为鲲鹏解决方案全景
2. 华为鲲鹏通用解决方案
- 3. 华为鲲鹏行业解决方案**
 - 政府行业解决方案
 - 金融行业解决方案
 - 运营商行业解决方案



基于鲲鹏计算平台，打造自主可控全栈政务云



政务OA



移动政务
简政



网格化管理



政务服务网



政务服务中心
惠民



信息共享



公共资源交易



经济运行监测
兴业

合作伙伴、集成商应用生态

基础云架构

PaaS

生态使能平台
集成商行业服务...

DaaS

数据湖
分布式数据存储、大数据分析...

IaaS

异构计算云
英特尔计算资源池、国产化计算资源池、存储池、网络/安全池

全面支持鲲鹏产品基础设施，可支持其他国产化平台

基础设施

服务器

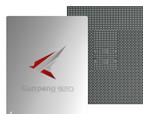
存储

网络

OS内核

基于鲲鹏处理器的全系列产品

基础芯片

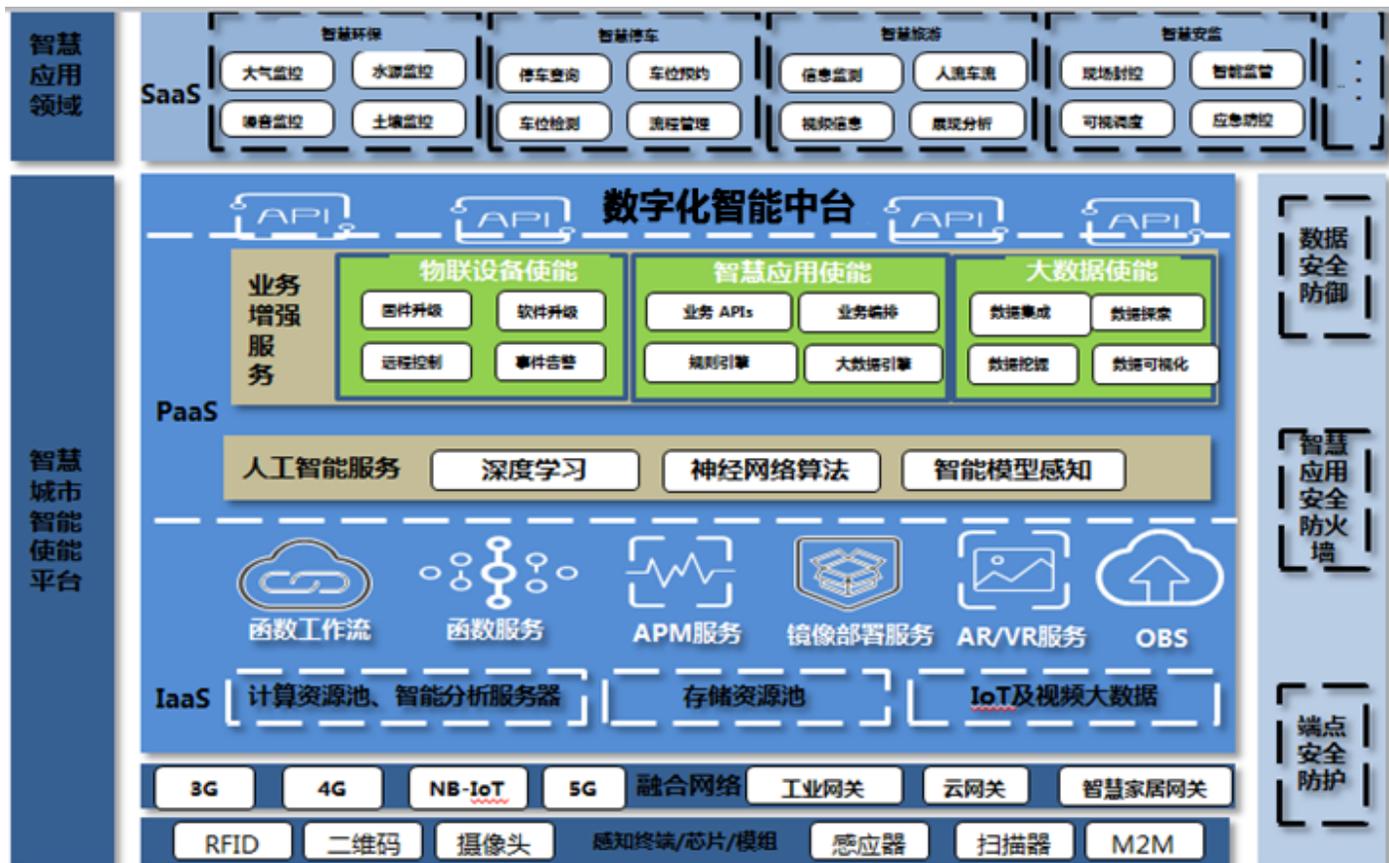


华为鲲鹏处理器 昇腾AI处理器 智能SSD控制器芯片 智能网卡芯片 智能管理芯片

构建关键部件鲲鹏高性能、高可靠基础设施（计算、存储、网络、管理）



软通智慧城市数字化智能中台



客户痛点及诉求

- 不能快速上云，对现有系统改动大
- API管理和开放能力低
- 基础设施运维工作量大
- 资源利用率低
- 打造行业级产品力量不足

解决方案优势

- 适应企业多种云化架构选择
- 弹性架构加速创新，海量容器秒级弹缩
- 开放的介入平台，支持镜像、服务、ISV、API接入
- 安全、可靠、高性能

Why Kungpeng?

- 国产自研芯片，芯片级别安全，自主可控
- 全栈鲲鹏云服务，支持各种组件，满足软件部署需求
- 华为云解决方案及研发支持快速响应





鲲鹏政务云

数字政府



公安



金融



交通



企业



教育



北京鲲鹏生态，引领全球商用

- 全球第一朵鲲鹏云：北京市政务云
- 首都之窗的提供商：首信科技自主可控云
- 祥云计划：北京市公安1000+TaiShan服务器，400T+Gauss DB数据库
- 全国电动汽车大数据中心：北理工数据中心
- 征信大数据平台：北京金控集团
-

首都之窗，思源政通-移动公服平台已完成入云测试



鹏城实验室 – 打造国家级开发者生态云平台

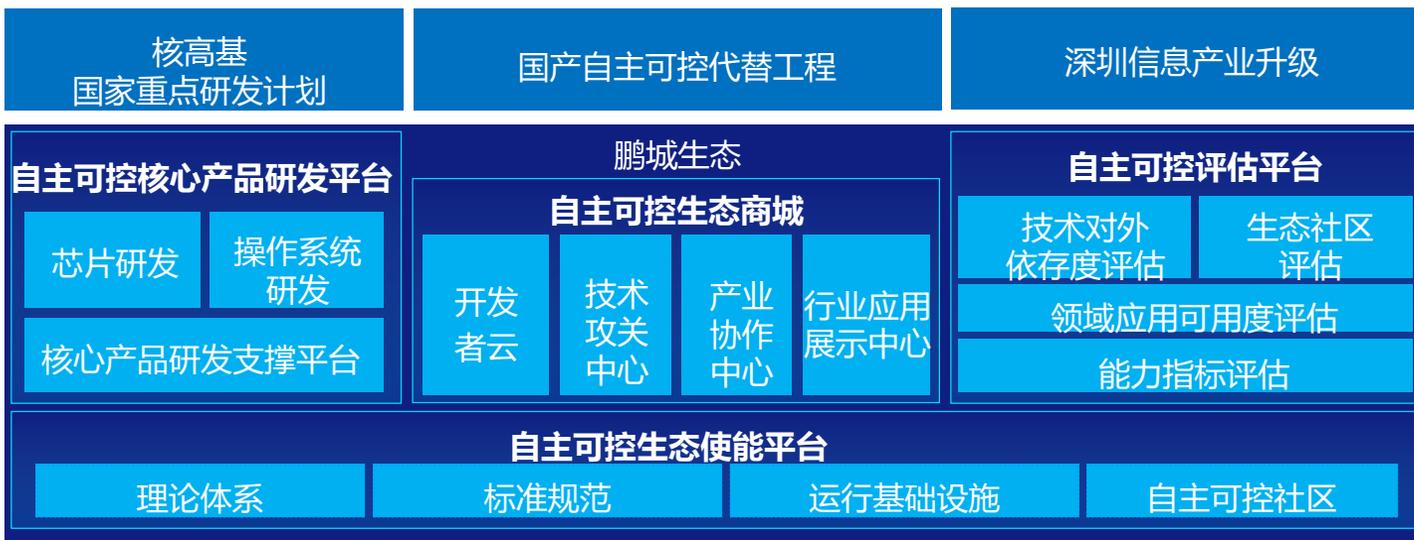


部署**150台TaiShan服务器**使能基础科研，生态开发

国家实验室“预备队”

具有全国影响力的科研机构

广东省实验室





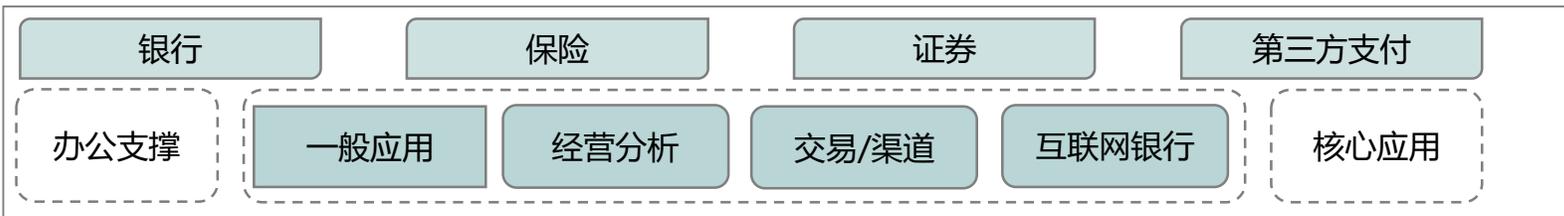
目录

1. 华为鲲鹏解决方案全景
2. 华为鲲鹏通用解决方案
- 3. 华为鲲鹏行业解决方案**
 - 政府行业解决方案
 - 金融行业解决方案
 - 运营商行业解决方案



基于鲲鹏计算平台：打造新一代智慧金融平台

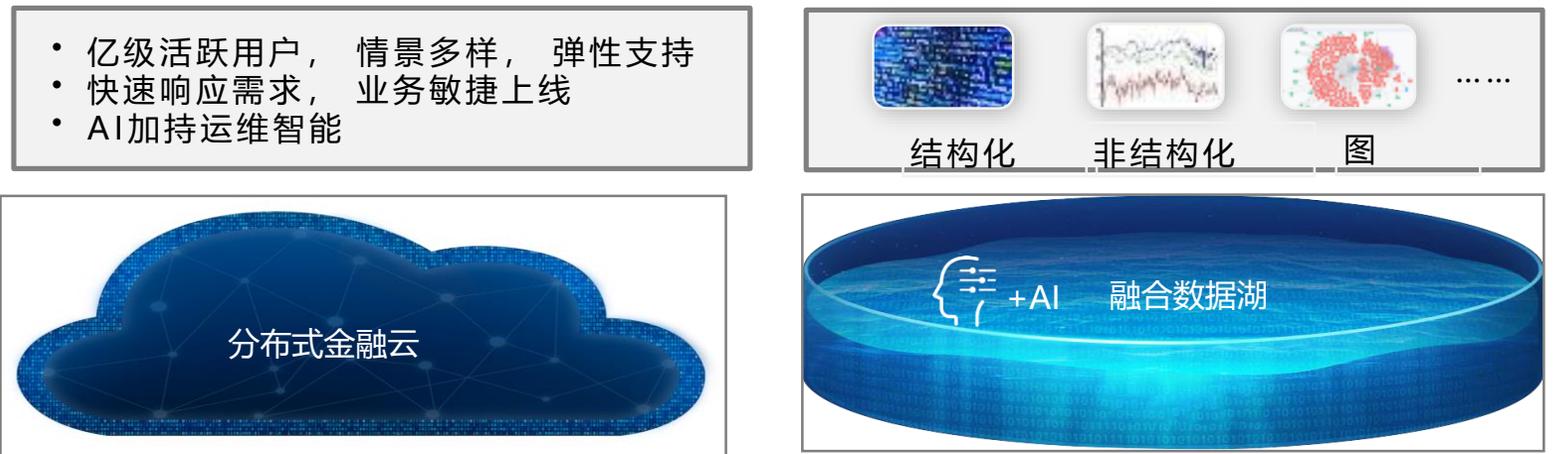
全场景金融



开放生态



全栈软件解决方案



智慧金融双底座，做大连接，做优体验

多源多模，海量数据实时处理

硬替代

办公支撑

互联网银行

经营分析

软替代

交易/渠道

准替代

核心应用



鲲鹏单集群支撑民生银行四大应用混合负载

支持多模引擎的HTAP数据库
HTAP database supports multimode engine



服务器1	服务器2	服务器3	服务器4	服务器5	服务器6
负载均衡			负载均衡		
协调节点	协调节点		协调节点		
备GTM	备GTM		主GTM		
备数据节点 2*备DN	备数据节点 1*备DN	备数据节点 1*备DN	备数据节点 2*备DN	备数据节点 1*备DN	备数据节点 1*备DN
				Keepalive 主节点	Keepalive 备节点
	主数据节点 1*主DN	主数据节点 1*主DN		主数据节点 1*主DN	主数据节点 1*主DN

多模引擎：支持5种类型数据混合负载

- 采用鲲鹏TaiShan服务器，组件分开部署，考虑每台服务器均衡压力
- 一套集群承载4个应用：贵金属模拟交易系统、员工福利一卡通系统、订单中心后管系统、企业家查询系统



与网联、银联共建第三方支付平台，支撑普惠金融



网联清算有限公司
NETSUNION CLEARING CORPORATION



- 分布式金融业务研发能力
- 开源软件业务应用能力
- 高并发金融业务场景
- 金融行业中立地位
- 过往业务中厂商产品试错经验

- 芯片级国产化产品研发能力
- 国产化基础设施产品能力（硬件、OS、云、数据。。。）
- 端到端自主可控方案能力
- 专业支撑团队

行业标
准制定

产品方
案创新

产业
创新

- **与网联共建第三方支付清算国产化平台**，打造金融极致高并发场景下的全栈自主创新解决方案。
- **与银联共建金融行业云**，基于华为混合云构建一致体验的服务能力，支撑建设国家普惠金融体系。



与重要金融机构，打造基于鲲鹏生态第二平面



信用卡核心业务下移/终端机具



经营分析数据库全量替换/大机下移联创



OLTP数据库替换



智慧监察大数据/办公系统

- 邮储银行
- 交通银行

- 华夏银行
- 民生银行

- 长沙银行
- 宁夏银行

- 太平保险
- 微众银行

- 厦门鲲鹏生态基地
- 成都鲲鹏生态基地



目录

1. 华为鲲鹏解决方案全景
2. 华为鲲鹏通用解决方案
- 3. 华为鲲鹏行业解决方案**
 - 政府行业解决方案
 - 金融行业解决方案
 - 运营商行业解决方案



ISV开发商均积极支持核心业务场景的国产化改造

开发语言	应用模块	技术难易分析
JAVA类应用	CRM: 前台WEB应用、缴费中心、产/商品中心、一级BOSS等, 占比52%; Billing: 客户信息接口、工单、账管界面等外网模块;	难度低, 无关键技术障碍, 主要是集成与性能验证
C/C++类应用	CRM: 业务受理、订单、渠道、统一接口平台等, 占比48%; Billing: 采集、计费、账务、结算等核心应用;	涉及改造 (char类型数据变量、浮点运算、内联汇编等) 与重编译 关键技术障碍: 访问oracle的应用依赖OCI



- 大数据、CRM系统改造已验证
- BOSS系统已完成改造



- CRM系统改造开始验证
- 计费系统已替换完成



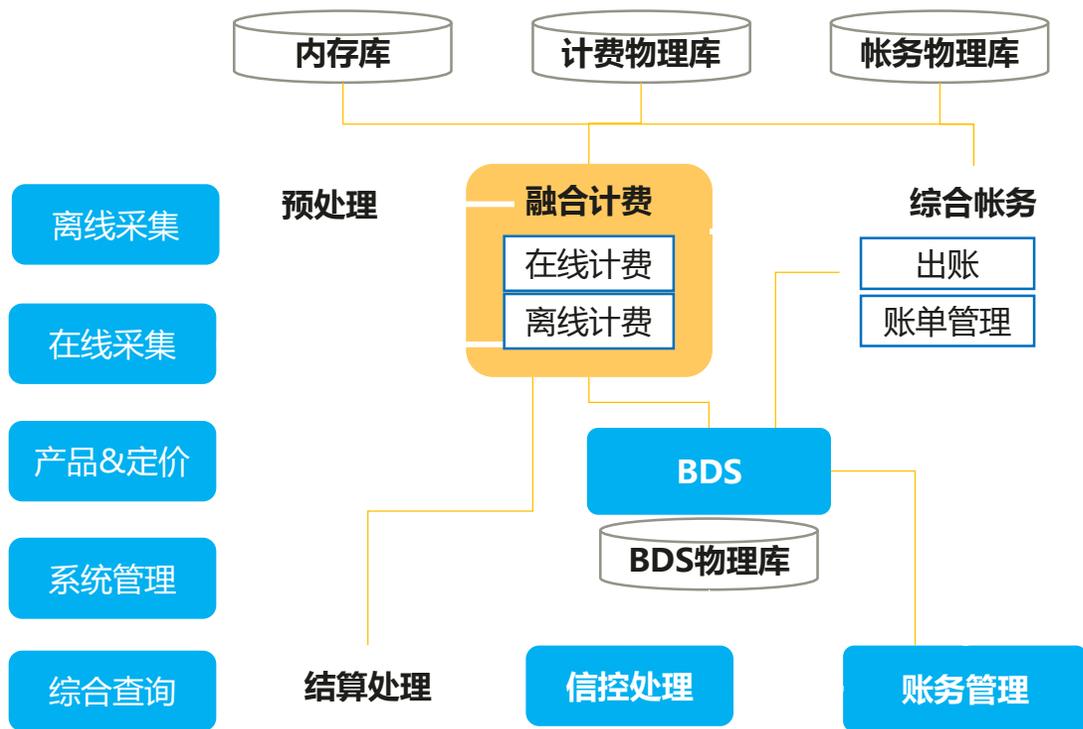
- 大数据系统改造已开始验证



- DPI系统改造已验证



计费系统改造



改造路径:

一阶段

- 1、移植JAVA类前台应用（图中蓝色模块），如产品定价、系统管理、综合查询等，采用鲲鹏虚拟机或容器部署
- 2、移植C/C++类的在线计费模块（黄色模块），基于国产内存库或文件方式处理，剥离C/C++中与Oracle 数据库（物理库）连接的代码并采用Java改造，解决鲲鹏不支持Oracle的问题

二阶段

- 1、移植C/C++类离线和批处理模块（白色模块），如离线采集、出账，结算等，采用GaussDB或国产数据库替代Oracle，并结合鲲鹏服务器多核优势进行优化性能

ISV改造进展:

亚信已经完成计费系统基于TaiShan服务器的迁移测试



面向5G，打造双平面云资源池

浙江移动于2019年6月29日成功打通全球首个基于鲲鹏的网络云VoLTE呼叫。

网络功能虚拟化编排器
(NFVO+)



鲲鹏网络
云资源池

基于鲲鹏的“中国芯”是
全国产化5G网络的里程碑

5GC IMS EPC 融合数据

混合组POOL



X86网络
云资源池

资源池独立部署：各部件无需考虑同时运行x86和鲲鹏，部署容易，运维简单

混合组POOL：x86、鲲鹏组成双平面，混合组POOL，保护已有投资



中国移动集中化大数据平台, 鲲鹏生态数据仓库验证

客户痛点：面向传统数据仓库减负和信息安全。

解决方案：现网场景验证基于鲲鹏的一体化MPP数据仓库。

Why Kunpeng：扩展性、可靠性、计算性能得到提升，具备可替换性。



痛点与挑战



数据仓库平台扩展性受限

现网MPP高可用、扩展性不足

替代方案总拥有成本高

难以满足未来信息安全需要

统一数据接入



自主可控的数据第二平面



思考题

1. 以下哪些属于鲲鹏通用解决方案？（ ）
 - A.HCSO 解决方案；
 - B.大数据解决方案；
 - C.HPC解决方案；
 - D.分布式存储；



本节小结

- 本章介绍了华为鲲鹏解决方案，包括通用解决方案及行业解决方案。



学习推荐

- <https://www.huaweicloud.com/kunpeng/solution.html>



谢谢

www.huawei.com



鲲鹏社区介绍



前言

- 本课程主要讲述鲲鹏社区是如何围绕鲲鹏体系来打造技术支持、知识共享和产业互助平台。
- 鲲鹏社区提供多元化云服务，基于云服务模式提供完整的自主可控软件栈，快速吸引新业务上线，提供多种业务系统迁移工具与指导等服务。
- 鲲鹏社区提供多样化的开发语言以及开发工具，建立完善的开发社区，培育体系化的职业发展方向，在开源社区及商业软件方面快速形成可支持全国产化技术体系的业务应用生态。



目标

- 学完本课程后，您将能够：
 - 熟悉鲲鹏社区的整体构架；
 - 了解鲲鹏社区各个板块的价值和功能；
 - 利用在鲲鹏社区中搜寻的资源实现自身鲲鹏相关业务的快速发展；



目录

1. 鲲鹏社区整体介绍
2. Hello,鲲鹏
3. 鲲鹏软件栈
4. 鲲鹏伙伴计划
5. 鲲鹏产品与解决方案
6. 鲲鹏学院
7. 鲲鹏论坛
8. 活动与资讯



鲲鹏社区

鲲鹏社区 —— 鲲鹏生态唯一综合型在线门户，提供完善的软件资源、技术知识、产品方案、生态政策、交易平台，汇聚全栈的资源 and 经验，使能鲲鹏开发者、合作伙伴技能成长，轻松完成应用移植和价值变现。鲲鹏社区链接如下：<https://www.huaweicloud.com/kunpeng/>



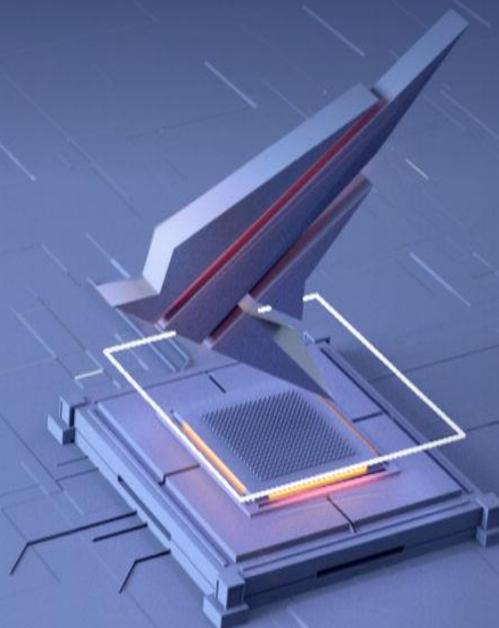


鲲鹏社区定位与使命

定位：鲲鹏社区是华为公司围绕鲲鹏技术体系打造的技术支持、知识共享和产业互助平台。

使命：全面支撑客户、合作伙伴和开发者，围绕鲲鹏技术体系开发有竞争力的计算产品和行业解决方案。

愿景：共建开放、共赢的鲲鹏生态，使能合作伙伴成功，共同做大鲲鹏产业。





加入鲲鹏社区，你可以获得



海量软件，简单快速完成开发移植

帮助合作伙伴、开发者简单快速地完成鲲鹏平台上的应用开发和移植



多维知识回馈，及时掌握鲲鹏最新技能

开放互助的知识论坛，使能合作伙伴、开发者提高技术能力



参与生态共建，共享产业发展机遇

开发者与华为公司共建鲲鹏产业生态，共享生态发展机遇



共建鲲鹏计算产业生态，共赢计算新时代

为了面向多样性计算时代，华为将携手产业合作伙伴一起构建鲲鹏计算产业生态，共同为各行各业提供基于华为鲲鹏处理器的领先IT基础设施及行业应用。华为将聚焦于鲲鹏和昇腾处理器、鲲鹏云服务和AI云服务等领域的技术创新，开放能力，使能伙伴，共同做大计算产业。

—— 立足中国，面向全球 ——

智慧政府 Bank 4.0 产业互联网 5G超高速互联 智能驾驶 智能制造 智慧医疗 智慧教育

大数据 分布式存储 数据库 原生应用 云服务 HPC MEC NFV 无人驾驶 机器人 Web...

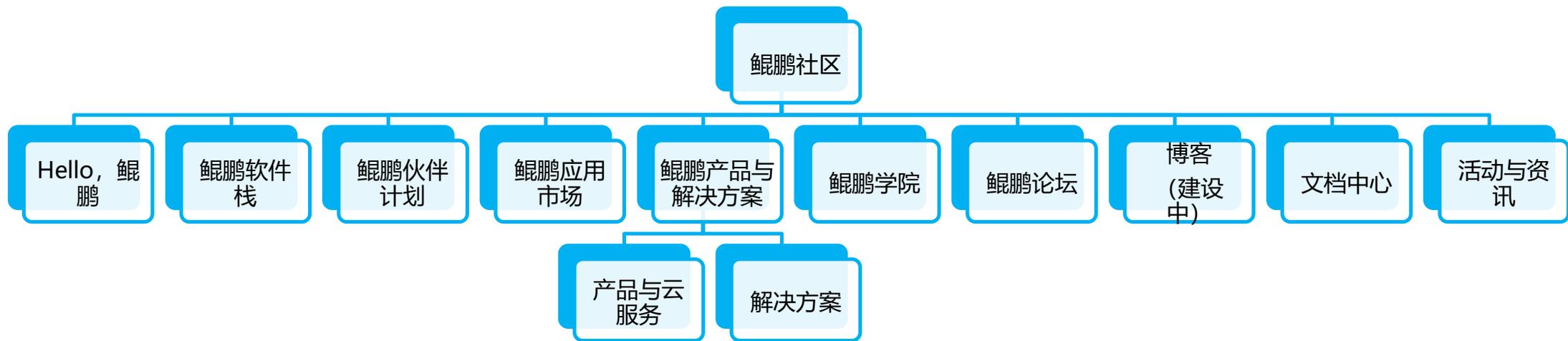
以行业聚合应用 以联盟孵化标准 以社区发展开发者

“鲲鹏 + 昇腾”芯片族



鲲鹏社区架构

鲲鹏社区作为一个面向开发者和合作伙伴的在线平台，为用户在线整合资源、知识、产品、政策、资讯、活动等鲲鹏生态内容，具体由以下板块构成：





目录

1. 鲲鹏社区整体介绍
2. **Hello,鲲鹏**
3. 鲲鹏软件栈
4. 鲲鹏凌云伙伴计划
5. 鲲鹏产品与解决方案
6. 鲲鹏学院
7. 鲲鹏论坛
8. 活动与资讯



Hello, 鲲鹏

Hello, 鲲鹏 是社区的一个信息汇聚入口，通过Hello, 鲲鹏，可以了解以下信息：



鲲鹏是什么

鲲鹏不仅仅局限于华为鲲鹏系列服务器芯片，更包含了兼容的服务器软件，以及建立在新计算架构上的完整软硬件生态和云计算生态。



提供什么价值

鲲鹏社区面向合作伙伴、开发者，为用户提供资源、知识、产品、政策、资讯、活动等鲲鹏生态内容，构建鲲鹏生态权威在线阵地。



怎么用鲲鹏

鲲鹏社区为合作伙伴和开发者提供软件包应用市场，加速商业变现，与鲲鹏社区一起 共赢计算新时代。



仅需三步，玩转鲲鹏软件

- 在Hello,鲲鹏 我们还可以了解到如何玩转鲲鹏软件：

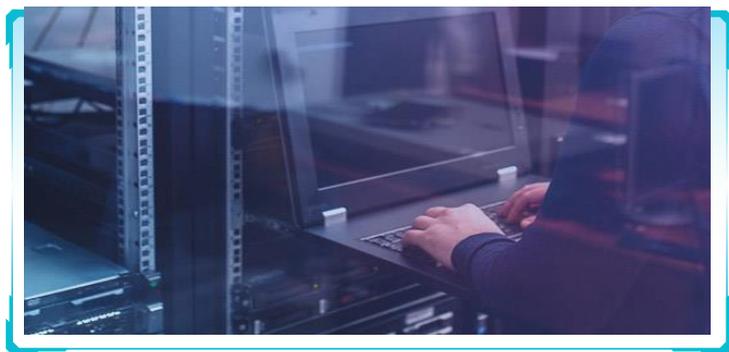
了解与评估



开发与测试



发布与部署



- 了解鲲鹏
- 获取常用软件移植指南
- 语言支持



- 环境构建
- 编译&运行
- 帮助&分享



- 镜像制作
- 加入鲲鹏应用市场
- 获取帮助



目录

1. 鲲鹏社区整体介绍
2. Hello,鲲鹏
- 3. 鲲鹏软件栈**
4. 鲲鹏凌云伙伴计划
5. 鲲鹏产品与解决方案
6. 鲲鹏学院
7. 鲲鹏论坛
8. 活动与资讯



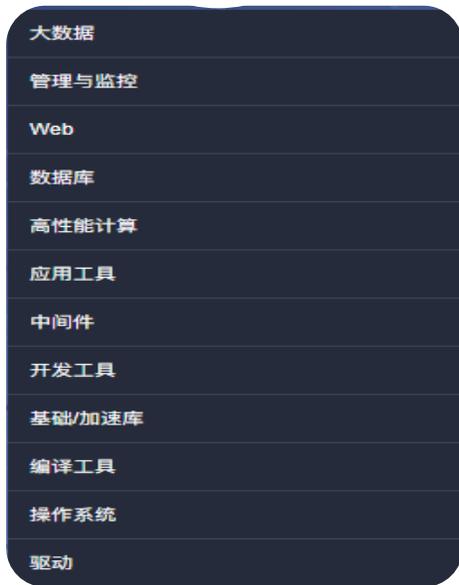
鲲鹏软件栈

鲲鹏软件栈是为开发者打造的一站式软件源获取平台，集华为自研、行业开源、开发者作品于一体，助力业务快速运行至鲲鹏平台。目前已上线200+主流软件。

5+
软件领域

11+
细分领域

200+
主流软件



CentOS 基于RHEL的Linux操作系统	Ubuntu 以桌面应用为主的开源GNU/Linux操作系统	中标麒麟 中标软件发布国产操作系统
深度OS 亦称为deepin, 旨在创建一个全新的简单、易用、美观的Linux操作系统	银河麒麟 中国自主知识产权的服务器操作系统	
Python 解释型、面向对象、动态数据类型的高级程序设计语言	PHP 通用开源脚本语言	Open JDK 开放源代码, 完全自由的开发环境
Ruby 面向对象程序设计的服务端脚本语言	GDB GNU开发组织发布的一个强大的UNIX下的程序调试工具	Lua 轻量小巧的脚本语言, 用标准C语言编写并以源代码形式开放
Hadoop 开源的分布式存储及计算框架	Hbase 高可靠性、高性能、面向列、可伸缩的分布式存储系统	Docker 开源的应用容器引擎
Kubernetes Google开源的容器编排管理系统	MySQL 关系型数据库管理系统	Nginx HTTP及反向代理服务器, 也可以提供邮件/FTP/UDP代理服务
Memcached 基于libevent的通用高性能分布式内存对象缓存系统	Redis 使用ANSI C语言编写的开源key-value内存数据库系统	Kafka 开源流处理平台, 用Scala和Java编写
RabbitMQ 实现了高级消息队列协议 (AMQP) 的开源消息代理软件 (亦称面向消息的中间件)	ZooKeeper 开放源码的分布式应用程序协调服务	ActiveMQ Apache软件基金会所研发的开源源代码消息中间件



如何获取软件

1、点击了解更多，进入
鲲鹏软件栈主页面



2、在鲲鹏软件中输入自
己想要获取的软件名称



3、点击软件名称即可获
取软件安装指南

鲲鹏软件栈

为您提供常用软件迁移和部署指南，助力业务快速运行至鲲鹏平台 [了解更多](#)

鲲鹏软件栈

了解如何将软件移植到鲲鹏上运行，获取操作指导和工具



Redis 高性能的key-value数据库

Predixy redis中间件

Hiredis Redis的C接口库

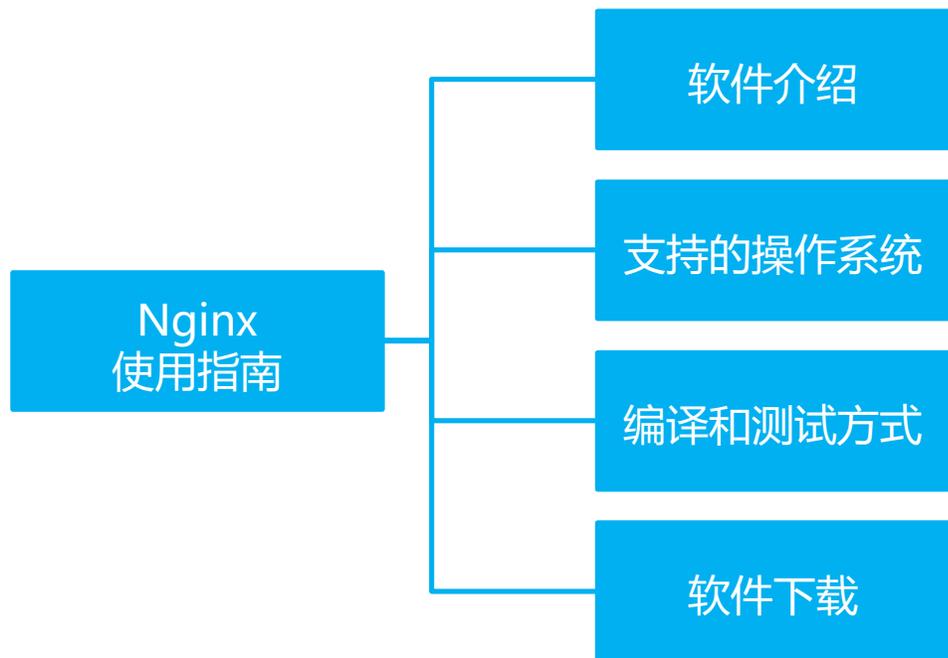
中间件 — Redis





软件使用指南内容构成

以Nginx为例，nginx是一个HTTP及反向代理服务器，也可以提供邮件/TCP/UDP代理服务。Nginx软件内容主要由以下几部分组成：



nginx
HTTP服务器、反向代理服务器
开发语言： C/C++

软件介绍

nginx是一个HTTP及反向代理服务器，也可以提供邮件/TCP/UDP代理服务。

支持的操作系统

经过华为云严格实测，以下操作系统在鲲鹏生态中可以完整运行nginx服务的全部功能：

- CentOS7.5
- Euler 2.8

编译和测试方式

软件下载

源码下载：[立即下载](#)



目录

1. 鲲鹏社区整体介绍
2. Hello,鲲鹏
3. 鲲鹏软件栈
- 4. 鲲鹏伙伴计划**
5. 鲲鹏产品与解决方案
6. 鲲鹏学院
7. 鲲鹏论坛
8. 活动与资讯



鲲鹏伙伴计划

鲲鹏伙伴计划是鲲鹏社区最全、最新、最官方的政策发布窗口，影响辐射鲲鹏生态万千开发者、合作伙伴；目前有鲲鹏凌云伙伴计划、鲲鹏展翅伙伴计划、鲲鹏智数伙伴计划等三大生态伙伴计划分别面向不同类型的开发者和合作伙伴。

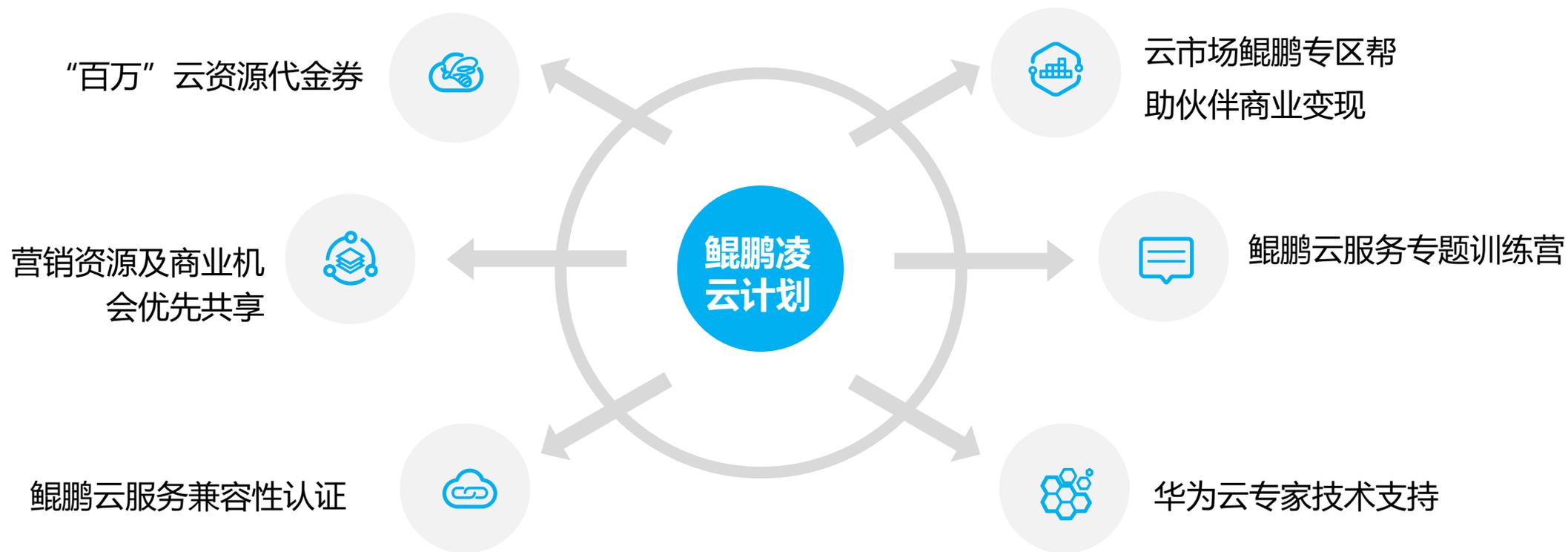
合作伙伴





鲲鹏凌云伙伴计划

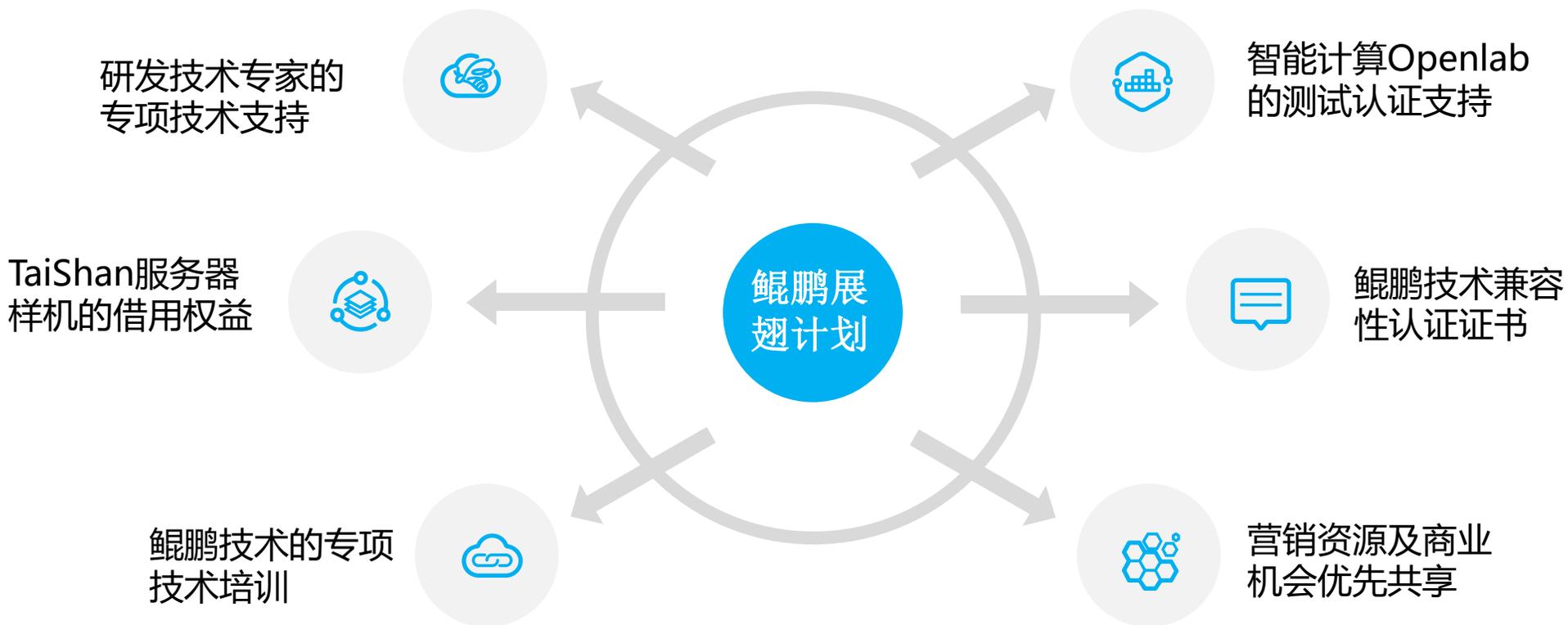
鲲鹏凌云伙伴计划是华为云推出的一项面向鲲鹏开放生态的合作伙伴计划。华为云为合作伙伴提供培训、技术、营销、市场的全方位支持，帮助伙伴基于华为鲲鹏云服务进行开发、应用移植，并开辟云市场鲲鹏专区，助力伙伴商业变现。





鲲鹏展翅伙伴计划

鲲鹏展翅伙伴计划是华为围绕TaiShan服务器推出的一项合作伙伴计划，旨在帮助更多的合作伙伴将应用迁移到TaiShan服务器上，共建鲲鹏生态，鲲鹏展翅伙伴计划为合作伙伴提供培训、技术、营销、市场的全面支持。



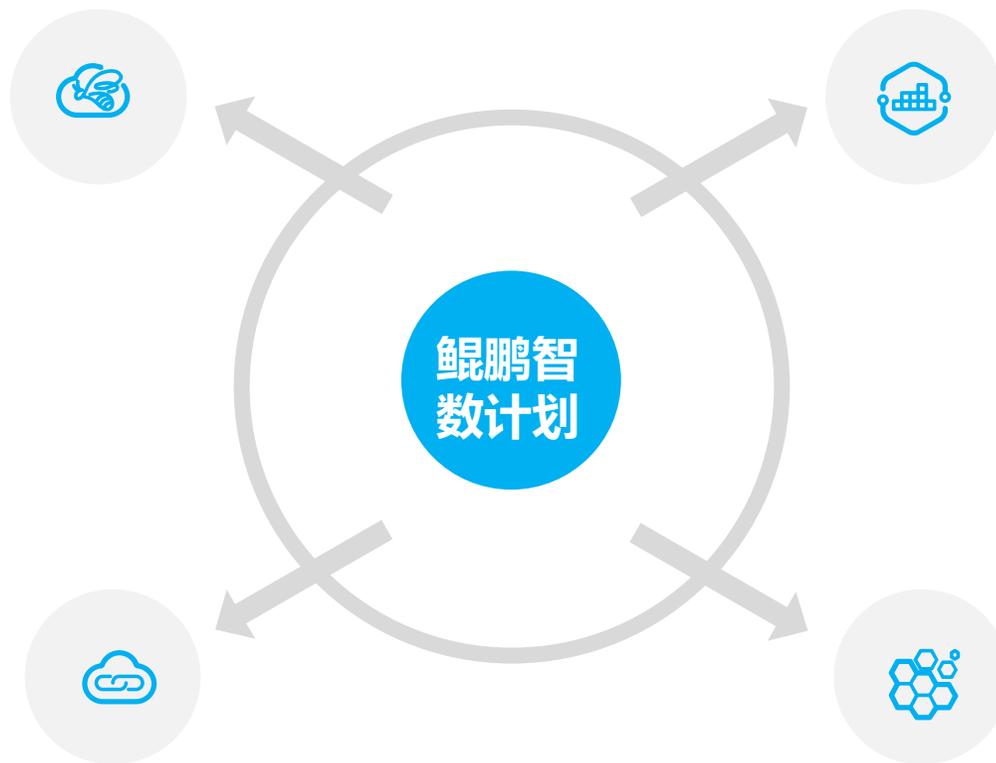


鲲鹏智数伙伴计划

鲲鹏智数伙伴计划是华为推出的一项面向智能数据产业，携手千行万业合作伙伴，共建开放鲲鹏生态的行动计划。为实现鲲鹏生态下智能数据产业繁荣，各类第三方软硬件产品与华为智能数据与存储解决方案互联互通，生态互助，商业成功，华为为合作伙伴提供全面技术与生态支持。

华为智能数据与存储产品
(存储、数据库、大数据
、桌面云等) 兼容性认证

智能数据与存储产品专题
训练营



智能数据与存储产品领域的
技术社区&专家技术支持

华为云专家技术支持



如何加入伙伴计划

以鲲鹏凌云伙伴计划为例，满足如下条件：有x86异构系统移植或开发能力、加入华为云解决方案伙伴计划或者华为云云市场计划；即可申请加入鲲鹏凌云伙伴计划，开启伙伴合作之旅。

伙伴合作旅程



鲲鹏凌云伙伴计划: <https://www.huaweicloud.com/partners/kunPeng/>

鲲鹏展翅伙伴计划: <http://ic-openlabs.huawei.com/#/teamdetail>

鲲鹏智数伙伴计划: <https://partner.huawei.com/>



目录

1. 鲲鹏社区整体介绍
2. Hello,鲲鹏
3. 鲲鹏软件栈
4. 鲲鹏伙伴计划
5. **鲲鹏产品与解决方案**
 - 产品与云服务
 - 解决方案
6. 鲲鹏学院
7. 鲲鹏论坛
8. 活动与资讯



鲲鹏产品与解决方案

鲲鹏产品与解决方案版块是鲲鹏社区产品与解决方案的主要发布窗口，主要介绍华为云鲲鹏生态云服务、TaiShan服务器及解决方案，丰富您的开发灵感。

鲲鹏产品与解决方案

了解并使用鲲鹏生态云服务、TaiShan服务器及解决方案，丰富您的开发灵感

产品与云服务

搭载算力强劲的华为鲲鹏处理器，为大数据、分布式存储、原生应用、高性能计算和数据库等应用高效加速，满足企业用户多样性计算、绿色计算的需求

[了解详情](#)

解决方案

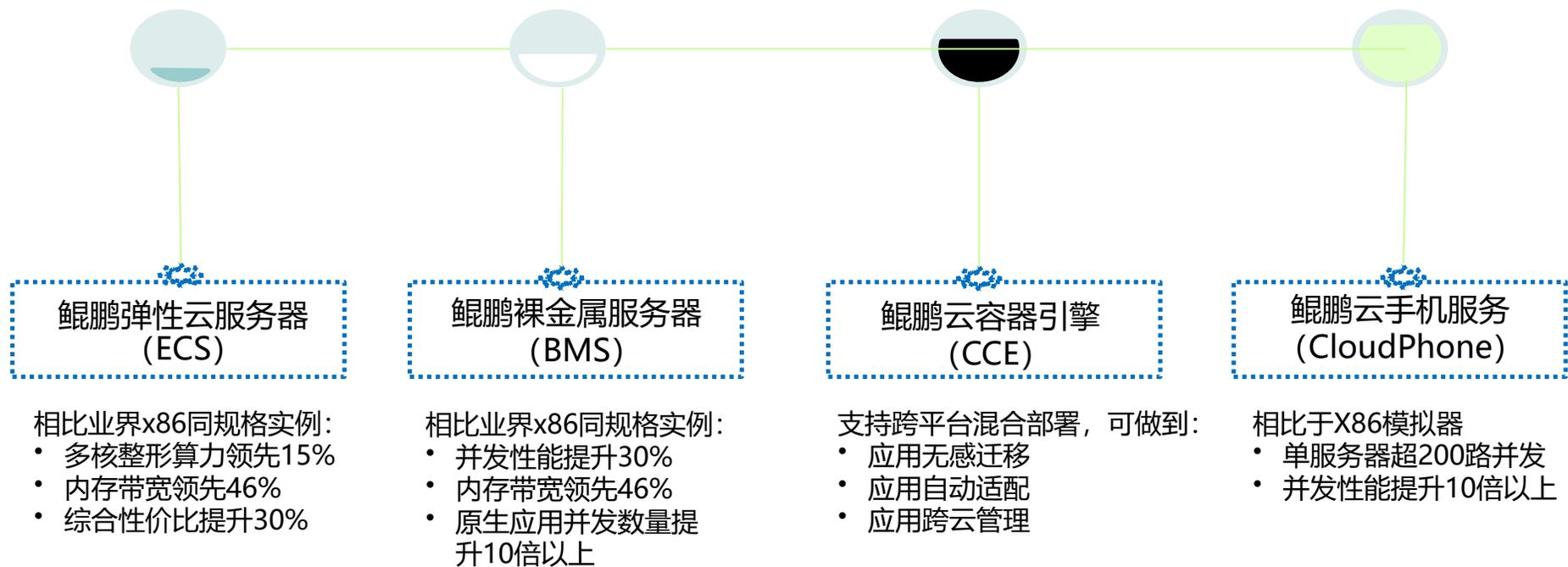
基于华为鲲鹏处理器和系列化芯片构筑的全栈IT基础设施解决方案，为政府、金融、运营商等重点行业提供端到端安全可信的解决方案，帮助客户推进基础设施集约化，业务应用全面云化和智能化

[了解详情](#)



华为云发布的鲲鹏云服务产品

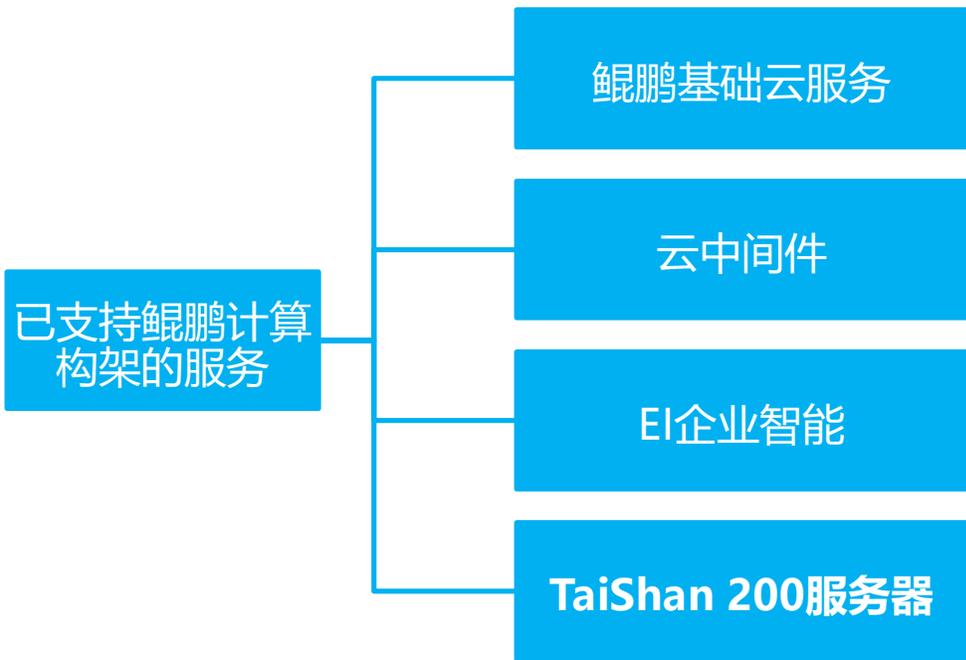
基于鲲鹏处理器，华为云推出鲲鹏云服务和解决方案，开启云上的多元新架构。华为云鲲鹏云服务具有同构、多核、全栈等能力，适用于不同场景，目前已正式发布69款基于鲲鹏处理器的云服务，后续还将陆续发布更多的云服务。





已支持鲲鹏计算构架的服务

目前华为云已支持鲲鹏计算构架的服务主要有以下类型：



鲲鹏基础云服务

已支持鲲鹏计算构架的基础云服务



弹性云服务器 ECS

全系机型覆盖，多核整形性能领先15%，综合性价比提升30%以上

[立即购买](#)



裸金属服务器 BMS

发挥鲲鹏多核架构优势，整机最高128核，并发性能提升30%，内存带宽提升46%

[立即购买](#)



云手机 CloudPhone

基于鲲鹏芯片的仿真手机服务，高度兼容手机应用，集成高性能GPU显卡，并发性能相比X86模拟器方案提升80%

[立即购买](#)

云中间件

已支持鲲鹏计算构架的云中间件



分布式缓存 Redis

首款自研ARM-Based的云缓存Redis服务，降低30%使用成本

[立即购买](#)



容器安全服务 CGS

扫描容器镜像中的漏洞，提供容器安全策略设置和防逃逸功能

[立即购买](#)



文档数据库服务 DDS

兼容MongoDB协议，提供一键部署，弹性扩容，容灾，备份，恢复，监控等服务能力

[立即购买](#)

EI企业智能

已支持鲲鹏计算构架的EI企业智能云服务



弹性大数据 MRS

众核为大数据提供高算力输出，性价比提升30%

[了解更多](#)



数据仓库服务 DWS

数据库查询性能提升26%，更加安全可靠

[立即购买](#)



云搜索服务 CSS

vCPU调度加速，查询性能提升15%

[立即购买](#)

TaiShan 200服务器

基于华为鲲鹏处理器的新一代数据中心服务器



2280均衡型

适合Web、大数据分析、软件定义存储等应用场景，可提供128核、2.6GHz主频的计算能力和最多16个NVMe SSD硬盘

[了解更多](#)



5280存储型

适合大数据分析、软件定义存储等应用场景，可提供128核、2.6GHz主频的计算能力和最高560TB本地存储容量

[了解更多](#)



X6000高密度型

适合高性能计算、互联网等应用场景的2U4节点高密服务器，可容纳4个2路计算节点

[了解更多](#)



鲲鹏产品与云服务优势

相比于其他同类型的产品和云服务，鲲鹏具有以下优势：

多核算力

- 多核整形算力领先15%
- 整机最高128核
- 整机并发性能提升30%

高性价比

- 计算性能领先8%
- 内存带宽领先46%
- 综合性价比领先30%

端云协同

- 原生应用场景，端云同构，性能提升80%
- 100%原生指令兼容，算法优化，无需改造

生态丰富

- 已兼容 CentOS、Ubuntu、EulerOS等20+款主流操作系统
- 100+企业核心业务应用，联合广大ISV，支持应用范围持续扩充



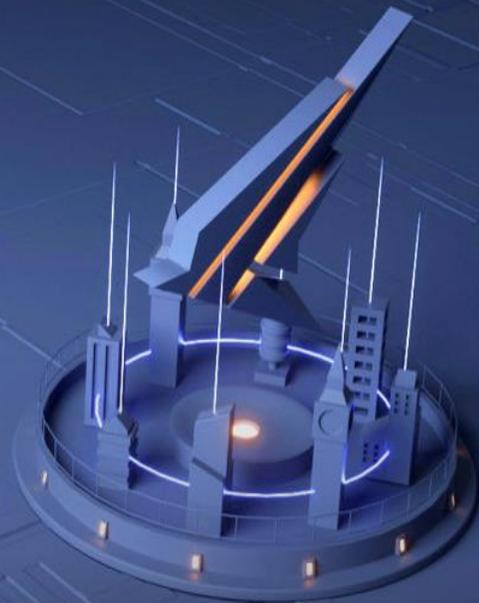
目录

1. 鲲鹏社区整体介绍
2. Hello,鲲鹏
3. 鲲鹏软件栈
4. 鲲鹏凌云伙伴计划
5. **鲲鹏产品与解决方案**
 - 产品与云服务
 - 解决方案
6. 鲲鹏学院
7. 鲲鹏论坛
8. 活动与资讯



华为云鲲鹏解决方案

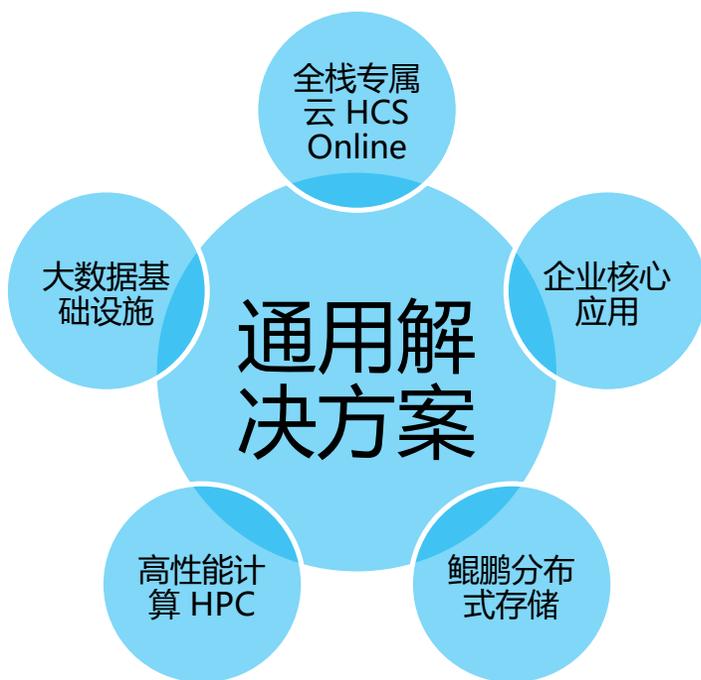
华为云鲲鹏解决方案是基于华为鲲鹏处理器和系列化芯片构筑的全栈IT基础设施解决方案，为政府、金融、运营商等重点行业提供端到端安全可信的解决方案，帮助客户推进基础设施集约化，业务应用全面云化和智能化。





华为云发布的鲲鹏行业解决方案

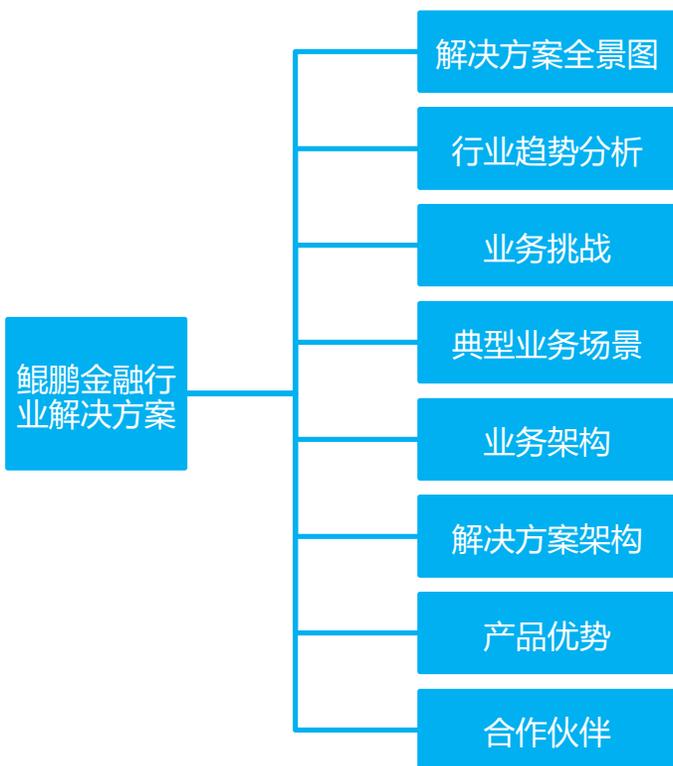
目前华为云已经发布的鲲鹏行业解决方案分为两大类：鲲鹏通用解决方案和鲲鹏行业解决方案。





样例：华为云鲲鹏金融行业解决方案

以金融行业为例：鲲鹏金融行业解决方案提供安全可靠的智慧金融全栈解决方案，助力金融机构实现个性化服务、智能化运营以及多元化生态，加速金融机构的数字化转型。





目录

1. 鲲鹏社区整体介绍
2. Hello,鲲鹏
3. 鲲鹏软件栈
4. 鲲鹏凌云伙伴计划
5. 鲲鹏产品与解决方案
6. **鲲鹏学院**
7. 鲲鹏论坛
8. 活动与资讯



鲲鹏学院

鲲鹏学院是鲲鹏生态的在线培训认证平台，体系化的培训课程，一站式体验学-练-证-训，快速完成学习覆盖，让用户轻松成为用鲲鹏产品进行开发、创新的专业人才。



学习

在线课程

华为云鲲鹏服务与解决方案
鲲鹏学习路径



实践

云端实验

鲲鹏云服务器部署Node.js项目
鲲鹏云服务器安装部署文字识别tesseract项目



认证

微认证

鲲鹏云服务器高可用性架构实践

职业认证

HCIA鲲鹏应用开发工程师认证



服务

线下培训

面授课程+上机实践
贴合实际+场景教学



鲲鹏在线课程

鲲鹏在线课程是鲲鹏社区为鲲鹏软件开发人员专业设计，旨在帮助开发人员了解鲲鹏体系的基础知识、软硬件产品、云服务，以及X86应用如何迁移上鲲鹏平台。目前主要提供以下课程：



鲲鹏系列课程

华为云鲲鹏服务与解决方案

TaiShan服务器及解决方案介绍

打造高效、可靠的TaiShan大数据平台

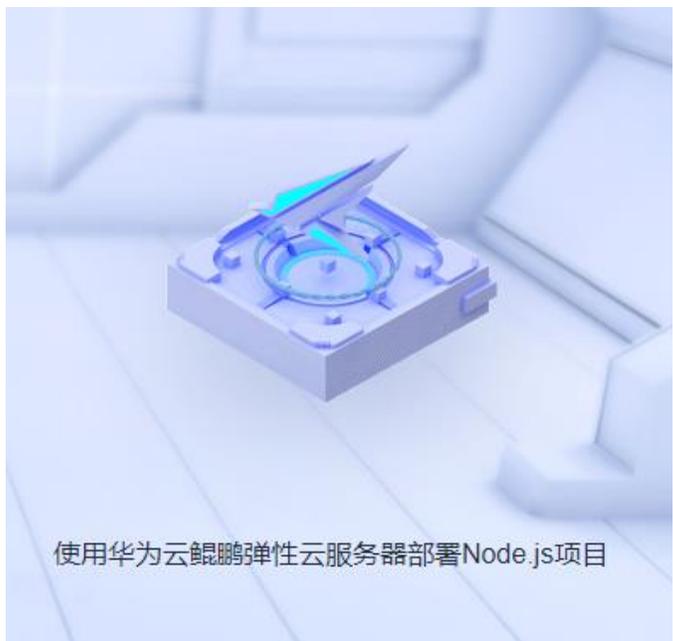
华为TaiShan 原生应用解决方案介绍

加速HPC与AI融合创新-TaiShan HPC解决方案介绍



鲲鹏云端实验室

鲲鹏云端实验室：使用虚拟华为云账号，根据详细的实验手册，一步步指导操作，模拟真实场景，完善的虚拟环境配置搭建，可快速体验华为云服务，在云端实现云服务的实践、调测和验证。



鲲鹏云端实验

使用华为云鲲鹏弹性云服务器部署Node.js项目

使用华为云鲲鹏弹性云服务器部署文字识别Tesseract项目

使用华为云鲲鹏弹性云服务器部署Discuz!项目



鲲鹏认证体系

鲲鹏认证体系是针对不同用户、产品类别，精心打造层次化的培训认证，目前提供HCIA职业认证和微认证。

职业认证可以帮助个人快速掌握鲲鹏相关理论知识和实践动手能力。

微认证则提供一站式在线学习、实验与考试，助力提升场景化技能。





目录

1. 鲲鹏社区整体介绍
2. Hello,鲲鹏
3. 鲲鹏软件栈
4. 鲲鹏凌云伙伴计划
5. 鲲鹏产品与解决方案
6. 鲲鹏学院
- 7. 鲲鹏论坛**
8. 活动与资讯



鲲鹏论坛

鲲鹏论坛是技术大拿、行业专家、热心开发者的交流圣地，各位名家高手坐而论道，共享实践经验、技术观点，目前拥有7大频道，500+干货主题，1000+讨论热帖，帮助开发者快速拓展技术视野。





目录

1. 鲲鹏社区整体介绍
2. Hello,鲲鹏
3. 鲲鹏软件栈
4. 鲲鹏凌云伙伴计划
5. 鲲鹏产品与解决方案
6. 鲲鹏学院
7. 鲲鹏论坛
8. **活动与资讯**



活动与资讯

了解鲲鹏生态一手资讯，参与热门活动，第一时间分享华为鲲鹏品牌及产品相关活动，华为鲲鹏芯片发布以及鲲鹏行业最新观点等资讯内容，为广大用户能更加深入的了解鲲鹏产业相关信息提供渠道，让用户可以时刻把握前沿科技脉搏。

活动与资讯

了解鲲鹏生态一手资讯，参与热门活动，时刻把握前沿科技脉搏 [热门活动](#) | [新闻资讯](#)



华为张顺茂：迎接拐点，拥抱计算新架构

移动应用的不断涌现催生了海量数据，人工智能的兴起带来了高密度计算。在这种情况下，企业必须...



化鲲为鹏，我有话说

超强福利来袭，发帖即可赢取荣耀手环5、2019华为全联接大会门票啦！



泰克实训云平台通过鲲鹏云服务兼容性认证

泰克教育通过参加华为云组织的鲲鹏云应用移植训练营，快速完成应用移植并成为华为云云市场严选计划伙伴



思考题

1. 以下哪些属于华为云鲲鹏伙伴计划？（ ）
 - A、鲲鹏凌云伙伴计划
 - B、鲲鹏智数伙伴计划
 - C、鲲鹏展翅伙伴计划
 - D、鲲鹏翱翔伙伴计划
2. 华为云鲲鹏社区目前是否提供软件下载功能？（ ）
 - A、是
 - B、不是



本章总结

- 本章简要介绍鲲鹏社区各个模块所承载的价值与信息所在，以及用户如何访问和使用社区工具的方式。帮助学员了解鲲鹏社区作为一个综合型在线门户，可以提供完善的软件资源、技术知识、产品方案、生态政策、交易平台等，汇聚全栈的资源和经验，使能鲲鹏开发者、合作伙伴技能成长，轻松完成应用移植和价值变现。



更多信息

相关链接:

鲲鹏社区首页: <https://www.huaweicloud.com/kunpeng/>

鲲鹏软件栈: <https://www.huaweicloud.com/kunpeng/software.html>

鲲鹏伙伴计划: <https://www.huaweicloud.com/kunpeng/partners.html>

鲲鹏产品与云服务: <https://www.huaweicloud.com/kunpeng/product.html>

鲲鹏解决方案: <https://www.huaweicloud.com/kunpeng/solution.html>

鲲鹏论坛: <https://bbs.huaweicloud.com/forum/forum-923-1.html>

鲲鹏计算产业: <https://e.huawei.com/cn/kunpeng>



谢谢

www.huawei.com