



3.1 鲲鹏软件栈

对于鲲鹏生态来说,最重要的是鲲鹏软件栈,也就是有哪些软件支持鲲鹏架构,设想一下,如果一个架构再优秀,硬件性能再强大,没有对应的软件支撑,也不会有人使用。所以,鲲鹏软件栈是否丰富,大家常用的软件是否可以在上面正常运行,就成了鲲鹏架构能否成功的关键。

鲲鹏架构目前适配的软件有几千种,每天还在持续增加,常用的开源软件、基础软件大部分都支持,下面从操作系统、数据库、中间件、Web 4 个维度列出一些适配好的软件并做一下简介,同时给出鲲鹏架构下最新适配的版本(除 3.1.1 节外,均指在 CentOS 操作系统下适配的最新版本)。

3.1.1 操作系统

1. CentOS

CentOS 是 Linux 发行版之一,它来自于 Red Hat Enterprise Linux(RHEL),依照开放源代码规定发布的源代码所编译而成。由于出自同样的源代码,因此有些要求高度稳定的服务器以 CentOS 替代商业版的 Red Hat Enterprise Linux 使用。两者的不同在于 CentOS 并不包含封闭源代码软件。CentOS 对上游代码的主要修改是为了移除不能自由使用的商标。

CentOS 官网网址为 <https://www.centos.org/>,编写本书时,鲲鹏架构支持的最新版本为 CentOS 8.1。

2. Ubuntu

Ubuntu 是以桌面应用为主的 Linux 发行版,也是目前用户最多的 Linux 版本,由英国 Canonical 有限公司主导开发和发布。

Ubuntu 官网网址为 <https://ubuntu.com/>,编写本书时,鲲鹏架构支持的最新版本为 Ubuntu 20.04。

3. 中标麒麟 OS

中标麒麟操作系统由中标软件有限公司开发,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。中标麒麟增强安全操作系统采用银河麒麟 KACF 强制访问控制框架和 RBA 角色权限管理机制,支持以模块化方式实现安全策略,提供多种访问控制策略的统一平台,是一款真正超越“多权分立”的 B2 级结构化保护操作系统产品。

中标麒麟 OS 官网网址: <http://www.cs2c.com.cn/>,编写本书时,鲲鹏架构支持的最新版本为 NeoKylin Server v7.0 U6。

4. 深度 OS

深度 OS 又称 Deepin,由武汉深之度科技有限公司开发,是一个基于 Linux 的操作系统,专注于提高使用者对日常办公、学习、生活和娱乐等操作的优异体验,适合笔记本、桌面计算机和一体机。Deepin 的历史可以追溯到 2004 年,其前身 Hiweed Linux 是中国第一个基于 Debian 的本地化衍生版,并提供轻量级的可用 Live CD,旨在创造一个全新的简单、易用、美观的 Linux 操作系统。

深度 OS 官网网址: <https://www.deepin.org/>,编写本书时,鲲鹏架构支持的最新版本为 UOS 20 Server。

5. openEuler

openEuler 是一个开源、免费的 Linux 发行版平台,通过开放的社区形式与全球的开发者共同构建一个开放、多元和架构包容的软件生态体系。同时,openEuler 也是一个创新的平台,鼓励任何人在该平台上提出新想法、开拓新思路、实践新方案。

openEuler 官网网址: <https://openeuler.org/>,编写本书时,鲲鹏架构支持的最新长期支持版本为 openEuler 20.03。

3.1.2 数据库

1. MySQL

MySQL 是一个高性能、低成本、可靠性好的开源数据库,被广泛地应用在网站和应用中,原开发者为瑞典的 MySQL AB 公司,该公司于 2008 年被 Sun 公司收购。2009 年,甲骨文公司收购 Sun 公司,MySQL 成为甲骨文公司旗下产品。

MySQL 官网网址: <https://www.MySQL.com/>,鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/MySQL0.html>,编写本书时,鲲鹏架构支持的最新版本为 MySQL 8.0。

2. MariaDB

MariaDB 数据库管理系统是 MySQL 的一个分支,由 MySQL 的创始人 Michael Widenius 主导开发,主要由开源社区维护,采用 GPL 授权许可。MariaDB 的目的是完全兼容 MySQL,包括 API 和命令行,使之能轻松成为 MySQL 的代替品。2008 年 2 月 26 日, Sun 微系统集团以大约 10 亿美元的价格收购了 MySQL AB,2009 年 Sun 公司被甲骨文公

司收购,MySQL 的原始开发者担心被甲骨文公司收购后 MySQL 是否能继续保持开源,于是在 2009 年 10 月 29 日发布了 MySQL 5.1 的复刻品 MariaDB 5.1。

MariaDB 官网网址: <https://mariadb.org/>, 鲲鹏架构安装文档网址: <https://www.huaweicloud.com/kunpeng/software/mariadb.html>, 编写本书时,鲲鹏架构支持的最新版本为 MariaDB 10.3.22。

3. PostgreSQL

PostgreSQL 是开源的对象-关系数据库管理系统,最初开始于加利福尼亚大学伯克利分校的 Ingres 计划。PostgreSQL 支持大部分 SQL 标准并且提供了很多其他现代特性,如复杂查询、外键、触发器、视图、事务完整性、多版本并发控制等,除此之外,PostgreSQL 允许用户定义基于正规 SQL 类型的新类型,允许数据库自身理解复杂数据,也允许类型继承。

PostgreSQL 官网网址: <https://www.postgresql.org/>, 鲲鹏架构迁移文档地址: <https://www.huaweicloud.com/kunpeng/software/postgresql.html>, 编写本书时,鲲鹏架构支持的最新版本为 PostgreSQL 11.3。

4. Cassandra

Cassandra 是一套开源分布式 NoSQL 数据库系统。它最初由 Facebook 开发,用于改善电子邮件系统搜索性能的简单格式数据,集 Google BigTable 的数据模型与 Amazon Dynamo 的完全分布式架构于一身。Facebook 于 2008 将 Cassandra 开源,此后,由于 Cassandra 良好的可扩展性和性能,被广泛采用,成为一种流行的分布式结构化数据存储方案,目前是 Apache 的顶级项目。

Cassandra 官网网址: <https://cassandra.apache.org/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/cassandra.html>, 编写本书时,鲲鹏架构支持的最新版本为 Cassandra 3.11.8。

5. MongoDB

MongoDB 是一种面向文档的数据库管理系统,用 C++ 等语言撰写而成,由 MongoDB 公司于 2007 年 10 月开发,2009 年 2 月首度推出,现以服务器端公共许可(SSPL)分发,社区版是免费的,可获得 Windows、Linux 和 OS X 系统的二进制版本。

MongoDB 官网网址: <https://www.mongodb.com/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/mongodb.html>, 编写本书时,鲲鹏架构支持的最新版本为 MongoDB 4.2.5。

3.1.3 中间件

1. Dubbo

Dubbo 是阿里巴巴公司开源的一款高性能、轻量级的 Java RPC 框架,可以和 Spring 框架无缝集成。它提供了三大核心能力:面向接口的远程方法调用,智能容错和负载均衡,以及服务自动注册和发现,目前是 Apache 的顶级项目。

Dubbo 官网网址: <http://dubbo.apache.org/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/dubbo.html>

www.huaweicloud.com/kunpeng/software/dubbo.html,编写本书时,鲲鹏架构支持的最新版本为 Dubbo 2.7.5。

2. Redis

Redis 是一个使用 ANSI C 编写的开源、支持网络、基于内存、可选持久性的键值对存储数据库,也是目前最流行的键值对存储数据库之一。从 2015 年 6 月开始,Redis 的开发由 Redis Labs 赞助。

Redis 官网网址: <https://redis.io/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/redis.html>,编写本书时,鲲鹏架构支持的最新版本为 Redis 6.0.2。

3. Kafka

Kafka 是由 Apache 软件基金会开发的一个开源流处理平台,由 Scala 和 Java 编写。该项目的目标是为处理实时数据提供一个统一、高吞吐、低延迟的平台。其持久化层本质上是一个“按照分布式事务日志架构的大规模发布/订阅消息队列”,这使它作为企业级基础设施来处理流式数据非常有价值。

Kafka 官网网址: <https://kafka.apache.org/>, 鲲鹏架构迁移文档网址: https://support.huaweicloud.com/prtg-apache-kunpengbds/kunpengbds_02_0008.html,编写本书时,鲲鹏架构支持的最新版本为 Kafka 2.11。

4. RabbitMQ

RabbitMQ 是一种实现了高级消息队列协议 (AMQP) 的开源消息代理软件。RabbitMQ 服务器是用 Erlang 语言编写的,而聚类和故障转移则构建在开放电信平台框架上。所有主要的编程语言均有与代理接口通信的客户端库。

RabbitMQ 官网网址: <https://www.rabbitmq.com/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/rabbitmq.html>,编写本书时,鲲鹏架构支持的最新版本为 RabbitMQ 3.7.27。

5. ZooKeeper

ZooKeeper 是 Apache 软件基金会的一个软件项目,它为大型分布式计算提供开源的分布式配置服务、同步服务和命名注册。ZooKeeper 的架构通过冗余服务实现高可用性。因此,如果第一次无应答,客户端就可以询问另一台 ZooKeeper 主机。

ZooKeeper 官网网址: <https://zookeeper.apache.org/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/zookeeper.html>,编写本书时,鲲鹏架构支持的最新版本为 ZooKeeper 3.6.2。

3.1.4 Web

1. Nginx

Nginx 是一款面向性能设计的免费开源 HTTP 服务器,根据类 BSD 许可证的条款发布,具有占有内存少、稳定性高等优势。Nginx 不采用每客户机一线程的设计模型,而是充

分使用异步逻辑,从而削减了上下文调度开销,所以并发服务能力更强。整体采用模块化设计,有丰富的模块库和第三方模块库,配置灵活,也可以用作反向代理、负载均衡器和 HTTP 缓存。

Nginx 官网网址: <http://Nginx.org/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/Nginx.html>, 编写本书时,鲲鹏架构支持的最新版本为 Nginx 1.19.1。

2. Apache

Apache HTTP Server(简称 Apache)是 Apache 软件基金会有一个开放源码的网页服务器软件,可以在大多数计算机操作系统中运行。由于其跨平台 and 安全性,被广泛使用,是最流行的 Web 服务器软件之一。它快速、可靠并且可通过简单的 API 扩展,将 Perl/Python 等解释器编译到服务器中。

Apache 官网网址: <http://httpd.apache.org/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/apache.html>, 编写本书时,鲲鹏架构支持的最新版本为 Apache HTTP Server 2.4.26。

3. Tomcat

Tomcat 是由 Apache 软件基金会属下 Jakarta 项目开发的 Servlet 容器,按照 Sun Microsystems 提供的技术规范,实现了对 Servlet 和 Java Server Page(JSP)的支持,并提供了作为 Web 服务器的一些特有功能,如 Tomcat 管理和控制平台、安全局管理和 Tomcat 阀等。由于 Tomcat 本身也内含了 HTTP 服务器,因此也可以视作单独的 Web 服务器。

Tomcat 官网网址: <https://tomcat.apache.org/>, 鲲鹏架构安装文档网址: <https://www.huaweicloud.com/kunpeng/software/tomcat.html>, 编写本书时,鲲鹏架构支持的最新版本为 Tomcat 9.0.13。

4. Node.js

Node.js 是能够在服务器端运行 JavaScript 的开放源代码、跨平台 JavaScript 运行环境。Node.js 采用 Google 开发的 V8 运行代码,使用事件驱动、非阻塞和异步输入输出模型等技术来提高性能,可优化应用程序的传输量和规模。这些技术通常用于资料密集的即时应用程序。

Node.js 官网网址: <https://Node.js.org/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/Node.js.html>, 编写本书时,鲲鹏架构支持的最新版本为 Node.js 13.14.0。

5. JBoss

JBoss 是一个基于 J2EE 的开放源代码的应用程序服务器,属于开源的企业级 Java 中间件软件。JBoss 代码遵循 LGPL 许可,可以在任何商业应用中免费使用。JBoss 是一个管理 EJB 的容器和服务器,支持 EJB 1.1、EJB 2.0 和 EJB3 的规范。2014 年 11 月 20 日,JBoss 更名为 WildFly。

WildFly 官网网址: <https://www.wildfly.org/>, 鲲鹏架构迁移文档网址: <https://www.huaweicloud.com/kunpeng/software/wildfly.html>

www.huaweicloud.com/kunpeng/software/jboss.html,编写本书时,鲲鹏架构支持的最新版本为 JBoss 17.0.0。

3.1.5 软件适配查询方式

如果要确定某种特定的软件是否已经适配鲲鹏架构,有两种查询方式,一种是在华为云查询,另一种是在华为计算开放实验室查询。

1. 华为云鲲鹏软件栈查询

华为云鲲鹏软件栈的官网网址为 <https://www.huaweicloud.com/kunpeng/software.html>,打开该网址,会出现鲲鹏软件栈查询页面,在输入框输入要查询的软件,会自动匹配出符合条件的软件列表,如图 3-1 所示。



图 3-1 鲲鹏软件栈

在下拉列表里单击具体的软件名称,例如 MySQL 5.7,会转向软件的适配详情页面,如图 3-2 所示。

可以在此页面找到详细的环境配置要求及进行适配操作需要的具体步骤。

2. 华为计算开放实验室查询

华为计算开放实验室的软件兼容性查询页面网址为 <http://ic-openlabs.huawei.com/openlab/#/unioncompaty>,在此可以按照开源软件、商业软件、操作系统 3 个类别进行软件的兼容性查询,查询的时候直接输入要查询的软件名称即可,以 Redis 为例,查询页面如图 3-3 所示。

单击“下载地址”列的链接,可以下载适配的软件;单击“编译指导”列的链接,可以给出具体的移植指南。

注意: 本节(3.1节)部分内容参考引用了维基百科,网址为 <https://zh.wikipedia.org/>,依据 CC BY-SA 3.0 许可证进行授权。要查看该许可证,可访问 <https://creativecommons.org/licenses/by-sa/3.0/>。



图 3-2 软件适配详情页

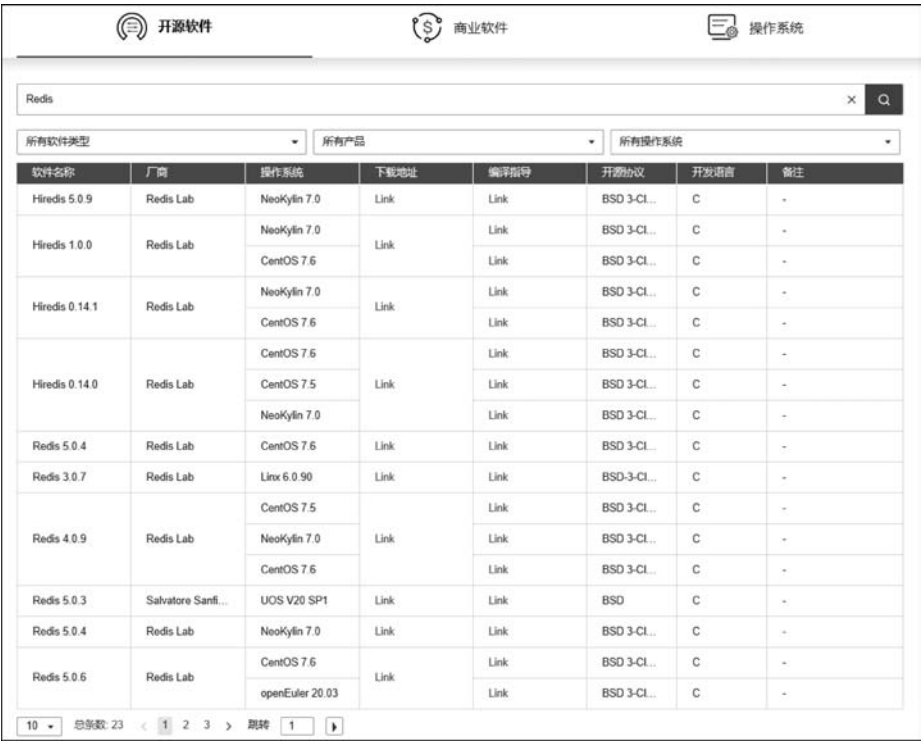


图 3-3 计算开放实验室软件查询

3.2 openEuler 操作系统

在 3.1 节介绍了鲲鹏软件栈,对适配的操作系统进行了简单描述,这里再着重介绍 openEuler 操作系统,因为该操作系统是华为根据鲲鹏架构的特点,在性能、可靠性、安全性等方面做了有针对性的优化,是为鲲鹏架构量身定制的操作系统。

3.2.1 openEuler 简介

openEuler 操作系统是华为推出的一款基于 Linux Kernel 4.19 版本的开源操作系统,该操作系统继承自 Linux Kernel 4.19 的部分执行 GNU GPL 第二版开源协议,华为自研的部分整体基于 Mulan PSL 协议。openEuler 操作系统版本分为两种,一种是社区创新版本,每 6 个月推出 1 个版本,另一种是 LTS 长期演进版本,每 2 年推出一个版本,目前有多个第三方厂商基于 LTS 版本发行了自己的商业发行版本,例如麒麟软件、普华、中科软等。openEuler 在硬件架构方面除了适配鲲鹏架构外,也兼容 x86 架构。

openEuler 官方网站网址为 <https://openeuler.org/>,可以在官网获取关于该操作系统的最新信息,截止到编写本书时,最新的社区创新版本是 2020 年 9 月发布的 openEuler 20.09,该版本生命周期为 6 个月,最新的 LTS 版本是 2020 年 3 月发布的 openEuler 20.03 LTS,该版本生命周期为 4 年。

作为开源的操作系统,openEuler 源码托管在 Gitee 平台,具有两个代码仓库,其中源码类项目存放网址为 <https://gitee.com/openeuler>,制作发布件所需的软件包存放网址为 <https://gitee.com/src-openeuler>。

3.2.2 关键特性

1. iSula 轻量级容器解决方案

openEuler 软件包中同时提供了 iSulad 与 Docker Engine 两种容器引擎,其中 iSula 轻量化通用容器引擎是一种新的容器解决方案,提供统一的架构设计来满足 CT 和 IT 领域的不同需求。相比 Golang 编写的 Docker,轻量级容器使用 C/C++ 实现,具有轻、灵、巧、快的特点,不受硬件规格和架构的限制,底层开销更小,可应用领域更为广泛,根据不同使用场景,提供多种容器形态,包括:

- (1) 适合大部分通用场景的普通容器。
- (2) 适合强隔离与多租户场景的安全容器。
- (3) 适合使用 systemd 管理容器内业务场景的系统容器。

iSulad 特性如下:

- (1) 缩短三级调用链,百容器内存资源占用相比 Docker 引擎显著下降。
- (2) 支持 CRI/OCI 标准开源接口,灵活对接 runc、kata 等多种 OCI 运行时。
- (3) 通过 Smart-loading 智能镜像下载技术,显著提升镜像下载速度。

(4) 安全容器：虚拟化技术和容器技术的有机结合，安全容器具有更好的隔离性。

(5) 系统容器：支持本地文件系统启动，可实现快速部署。支持部署 systemd，提升 user namespace 隔离性。

2. Kunpeng 加速引擎(KAE)，支持加解密加速

支持的主要算法如下：

(1) 摘要算法 SM3，支持异步模型。

(2) 对称加密算法 SM4，支持异步模型，支持 CTR/XTS/CBC 模式。

(3) 对称加密算法 AES，支持异步模型，支持 ECB/CTR/XTS/CBC 模式。

(4) 非对称算法 RSA，支持异步模型，支持 Key Sizes 1024/2048/3072/4096。

(5) 密钥协商算法 DH，支持异步模型，支持 Key Sizes 768/1024/1536/2048/3072/4096。

3. A-Tune 智能系统性能优化引擎

操作系统作为衔接应用和硬件的基础软件，如何调整系统和应用配置，充分发挥软硬件能力，从而使业务性能达到最优，对用户至关重要。然而，运行在操作系统上的业务类型成百上千，应用形态千差万别，对资源的要求各不相同，随着业务复杂度和调优对象的增加，调优所需的时间成本呈指数级增长，导致调优效率急剧下降，调优成为一项极其复杂的工程，给用户带来巨大挑战。

其次，操作系统作为基础设施软件，提供了大量的软硬件管理能力，每种能力适用场景不尽相同，并非对所有的应用场景都通用有益，因此，不同的场景需要开启或关闭不同的能力，组合使用系统提供的各种能力，才能发挥应用程序的最佳性能。

为了应对上述挑战，openEuler 推出了 A-Tune。A-Tune 是一款基于 AI 开发的系统性能优化引擎，它利用人工智能技术，对业务场景建立精准的系统画像，感知并推理出业务特征，进而做出智能决策，匹配并推荐最佳的系统参数配置组合，使业务处于最佳运行状态。

4. 增强 glibc/zlib/gzip 性能

充分利用 AArch 64 的 neon 指令集，提升基础库性能。

5. 内核特性增强

(1) 支持 ARM 64 内核热补丁。

(2) Numa Aware Qspinlock：减少跨 NUMA 节点的 Cache/总线冲突。

(3) 通过优化 IOVA 页表查找和页表释放算法，提升 I/O MMU 子系统性能。

(4) 根据 ARM 64 指令及流水线特点，优化 CRC32 及 checksum 实现，大幅提升数据校验性能。

(5) 支持 ARM v8.4 MPAM(Memory System Resource Partitioning and Monitoring)。

6. 虚拟化特性增强

(1) 中断虚拟化优化：IRQfd 路径注入中断优化，大幅提升高性能直通设备(网卡、SSD 磁盘等)性能。

(2) 内存虚拟化优化：借助鲲鹏硬件特性，提升虚拟机启动内存加载速度。

(3) 存储虚拟化优化：iSCSI 模块 kworker 的 NUMA 亲和性自绑定优化，提升 IPSAN 磁盘的 I/O 性能。

注意：本节(3.2.2 节关键特性)内容参考引用了 openEuler 官方文档，网址为 <https://openEuler.org/zh/documentation>，依据 CC BY-SA 4.0 许可证进行授权。要查看该许可证，可访问 <https://creativecommons.org/licenses/by-sa/4.0/>。

3.2.3 操作系统命令

openEuler 的命令和主流的 Linux 操作系统的命令是兼容的，本书后续章节会大量使用命令，这里重点介绍常用的操作命令，基本上覆盖了后续使用的需要，如果读者对 Linux 很熟悉，可以跳过本节。

本节在介绍操作系统命令时，不会详细描述某一个命令的所有参数，只是介绍最常用的用法，目的是让读者快速了解常用命令的使用，更全面的用法可以参考专门的操作系统书籍。

1. 常用系统命令

(1) shutdown：关机，根据参数不同可以立即关机或者定时关机。

如果没有参数，则默认 1min 后关机，代码如下：

```
shutdown
```

如果参数为 now，则表示立刻关机，代码如下：

```
shutdown now
```

如果参数为时间，则表示计划关机的时间，下面的命令表示 9:00:00 关机，代码如下：

```
shutdown 9:00
```

如果参数为-c，则表示取消关机计划，代码如下：

```
shutdown -c
```

(2) reboot：重新启动操作系统。

(3) poweroff：关机。

(4) exit：退出 shell，关闭当前终端。

(5) lscpu：列出 CPU。一般使用该命令确认当前服务器的架构，对于鲲鹏架构的服务器，该命令及回显如下：

```
[root@ecs - kunpeng ~]# lscpu
Architecture:          aarch64
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):         1
Model:                 0
CPU max MHz:           2400.0000
CPU min MHz:           2400.0000
BogoMIPS:              200.00
L1d cache:             64K
L1i cache:             64K
L2 cache:              512K
L3 cache:              32768K
NUMA node0 CPU(s):    0-3
Flags:                 fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid
asimdrdm jscvt fcma dcpop asimddp asimdfhm
```

(6) free: 查看内存。

(7) top: 查看系统资源实时信息,这是一个常用的调试辅助指令,可以确认哪些进程在使用资源、命令及反馈,查询结果如下:

```
[root@ecs - kunpeng ~]# top
top - 08:48:25 up 6 min, 1 user, load average: 0.00, 0.13, 0.09
Tasks: 123 total, 2 running, 84 sleeping, 0 stopped, 0 zombie
% Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 32890752 total, 32214976 free, 347008 used, 328768 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 29791872 avail Mem

  PID USER      PR    NI    VIRT    RES    SHR S  % CPU  % MEM   TIME+  COMMAND
    1 root        20     0   156928   8576   4416 S   0.0   0.0   0:01.45 systemd
    2 root        20     0         0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root         0   -20         0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root         0   -20         0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root         0   -20         0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kb
    7 root        20     0         0      0      0 I   0.0   0.0   0:00.01 kworker/u8:0-fl
    8 root         0   -20         0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root        20     0         0      0      0 S   0.0   0.0   0:00.00 ksoftirqd/0
   10 root        20     0         0      0      0 I   0.0   0.0   0:00.01 rcu_sched
   11 root        rt      0         0      0      0 S   0.0   0.0   0:00.07 migration/0
   12 root        rt      0         0      0      0 S   0.0   0.0   0:00.00 watchdog/0
   13 root        20     0         0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
```

```

14 root 20      0      0 0 0 S  0.0  0.0  0:00.00 cpuhp/1
15 root rt      0      0 0 0 S  0.0  0.0  0:00.00 watchdog/1
16 root rt      0      0 0 0 S  0.0  0.0  0:00.07 migration/1
17 root 20      0      0 0 0 S  0.0  0.0  0:00.00 ksoftirqd/1
19 root 0      -20      0 0 0 I  0.0  0.0  0:00.00 kworker/1:0H-kb
20 root 20      0      0 0 0 S  0.0  0.0  0:00.00 cpuhp/2
21 root rt      0      0 0 0 S  0.0  0.0  0:00.00 watchdog/2
22 root rt      0      0 0 0 S  0.0  0.0  0:00.07 migration/2
23 root 20      0      0 0 0 S  0.0  0.0  0:00.00 ksoftirqd/2
25 root 0      -20      0 0 0 I  0.0  0.0  0:00.00 kworker/2:0H-kb
26 root 20      0      0 0 0 S  0.0  0.0  0:00.00 cpuhp/3
27 root rt      0      0 0 0 S  0.0  0.0  0:00.00 watchdog/3
28 root rt      0      0 0 0 S  0.0  0.0  0:00.07 migration/3
29 root 20      0      0 0 0 S  0.0  0.0  0:00.00 ksoftirqd/3
31 root 0      -20      0 0 0 I  0.0  0.0  0:00.00 kworker/3:0H-kb
32 root 20      0      0 0 0 S  0.0  0.0  0:00.00 kdevtmpfs
33 root 0      -20      0 0 0 I  0.0  0.0  0:00.00 netns
34 root 20      0      0 0 0 S  0.0  0.0  0:00.00 kauditd
35 root 20      0      0 0 0 I  0.0  0.0  0:00.00 kworker/0:1-eve
37 root 20      0      0 0 0 S  0.0  0.0  0:00.00 khungtaskd
38 root 20      0      0 0 0 S  0.0  0.0  0:00.00 oom_reaper
39 root 0      -20      0 0 0 I  0.0  0.0  0:00.00 writeback
40 root 20      0      0 0 0 S  0.0  0.0  0:00.00 kcompactd0
41 root 25       5      0 0 0 S  0.0  0.0  0:00.00 ksmd
42 root 39     19      0 0 0 S  0.0  0.0  0:00.00 khugepaged
43 root 0      -20      0 0 0 I  0.0  0.0  0:00.00 crypto
44 root 0      -20      0 0 0 I  0.0  0.0  0:00.00 kintegrityd
45 root 0      -20      0 0 0 I  0.0  0.0  0:00.00 kblockd

```

在 top 状态下继续按“1”，可以列出每个 CPU 核心的使用信息，演示服务器包含 4 个核心，所以会列出 Cpu0~Cpu3 的相信信息：

```

top - 08:51:03 up 9 min, 1 user, load average: 0.00, 0.07, 0.08
Tasks: 125 total, 1 running, 84 sleeping, 0 stopped, 0 zombie
% Cpu0 : 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
% Cpu1 : 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
% Cpu2 : 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
% Cpu3 : 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 32890752 total, 32214080 free, 347072 used, 329600 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 29791296 avail Mem

```

在 top 状态下继续按 M 键，可以切换内存详细信息的显示。

在 top 状态下按 Shift+M 键，可以按照内存占用率大小按顺序排列进程列表。

2. 文件与目录命令

(1) cd：切换目录命令。

切换到主目录,命令如下:

```
cd ~
```

切换到根目录下的 etc 目录,命令如下:

```
cd /etc
```

切换到当前目录下的 subdir 目录,命令如下:

```
cd subdir
```

切换到根目录,命令如下:

```
cd /
```

切换到上 1 级目录,命令如下:

```
cd ..
```

切换到上 2 级目录,命令如下:

```
cd ../..
```

(2) mkdir: 创建目录命令。

在目录 opt 下创建 data 目录,命令如下:

```
mkdir /opt/data/
```

递归创建目录,如果最底层的 c 目录的任何一个父目录不存在,则创建它,命令如下:

```
mkdir -p /opt/a/b/c
```

(3) touch: 创建空白文件命令。

在/opt/data 目录下创建文件 test.conf,命令如下:

```
touch /opt/data/test.conf
```

(4) rm: 删除文件或文件夹命令。

删除/opt/data 目录下的 test.conf 文件,删除时需要确认,输入 y 便可以删除,否则放弃删除文件,命令如下:

```
rm /opt/data/test.conf
```

直接删除/opt/data 目录下的 test.conf 文件,不询问是否删除,命令如下:

```
rm -f /opt/data/test.conf
```

直接删除/opt/data 目录及目录下的所有文件,不询问是否删除,命令如下:

```
rm -rf /opt/data/
```

删除/opt/data 目录下所有扩展名为.conf 的文件,删除时逐个确认是否删除,命令如下:

```
rm /opt/data/*.conf
```

(5) ls: 列出目录命令。

列出当前目录,命令如下:

```
ls
```

列出所有的文件,包括隐藏文件,命令如下:

```
ls -a
```

列出文件时显示详细信息,命令如下:

```
ls -l
```

(6) pwd: 显示工作路径。

(7) mv: 移动文件或目录命令。

文件重命名,把文件 a.conf 重命名为 b.conf,命令如下:

```
mv a.conf b.conf
```

移动文件,把文件/opt/data/目录下的 a.conf 文件移动到/opt/目录下,命令如下:

```
mv /opt/data/a.conf /opt/
```

移动目录,把/opt/data/目录移动到/tmp/目录下面,命令如下:

```
mv /opt/data/ /tmp/
```

(8) cp: 复制文件或目录命令。

复制文件,把/opt/目录下的 b.conf 文件复制到/tmp/data/目录下,命令如下:

```
cp /opt/b.conf /tmp/data/
```

复制目录,把/tmp/data/目录复制到/opt/目录下,命令如下:

```
cp -r /tmp/data/ /opt
```

(9) find: 查找文件命令。

在/opt/目录查找所有扩展名为.conf的文件,命令如下:

```
find /opt/ -name '*.conf'
```

3. 文件查看编辑命令

(1) cat: 查看文件内容命令。

查看环境变量文件,此时会显示整个文件内容,命令如下:

```
cat /etc/profile
```

(2) more: 分页查看文件内容命令。

查看环境变量文件,分页显示文件内容,命令如下:

```
more /etc/profile
```

(3) head: 查看文件开头部分内容命令。

查看环境变量文件前3行内容,命令如下:

```
head -3 /etc/profile
```

(4) tail: 查看文件尾部内容命令。

查看环境变量文件最后3行内容,命令如下:

```
tail -3 /etc/profile
```

(5) vi: 创建或编辑文件内容命令。

如果/opt/data/a.conf不存在,则创建并编辑该文件。如果存在此文件,则直接编辑该文件。vi共分为3种模式,分别是命令模式、输入模式和底线命令模式,命令如下:

```
vi /opt/data/a.conf
```

■ 命令模式

刚启动 vi 进入命令模式,在此模式下输入的字符被当作命令,常用的命令如下:

i: 切换到输入模式。

x: 删除当前光标所在处的字符。

: 切换到底线命令模式,可以在最后一行输入命令。

■ 输入模式

输入模式可以进行正常输入,按 Esc 键退出输入模式,进入命令模式。

■ 底线命令模式

底线命令模式下的基本命令如下:

q: 退出程序。

w: 保存文件。

wq: 保存修改并退出。

q!: 强制退出但不保存。

4. 软件包管理命令

在 openEuler 操作系统中,软件包管理工具有两种,一种是首选的 DNF,另一种是被广泛使用了很多年的 YUM。DNF 和 YUM 的大部分命令是兼容的,只要熟悉了一种工具的使用,基本上可以按照同样的命令使用另一种工具。在后续的章节里,所使用的操作系统主要是 CentOS 7,但是 CentOS 7 默认是不安装 DNF 的,所以这里使用 YUM 工具来演示软件包管理命令。

(1) yum search: 在 rpm 仓库中搜寻软件包,用法如下:

```
yum search ftp
```

该命令会从软件仓库查找所有包含 ftp 的软件包,并显示匹配软件包的详细信息,命令及回显如下(回显内容过多,这里只显示部分内容):

```
[root@ecs-kunpeng ~]# yum search ftp
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirrors.huaweicloud.com
* epel: mirrors.tuna.tsinghua.edu.cn
* extras: mirrors.huaweicloud.com
* updates: mirrors.huaweicloud.com
===== N/S matched: ftp =====
... 此处省略大部分内容
tnftp.aarch64 : FTP (File Transfer Protocol) client from NetBSD
uberftp.aarch64 : GridFTP-enabled ftp client
vsftpd.aarch64 : Very Secure Ftp Daemon
vsftpd-sysvinit.aarch64 : SysV initscript for vsftpd daemon
cURL.aarch64 : A utility for getting files from remote servers (FTP, HTTP, and others)
```

```

debmirror.noarch : Debian partial mirror script, with ftp and package pool support
dmlite-dpm-dsi.aarch64 : Disk Pool Manager (DPM) plugin for the Globus GridFTP server
dpm-dsi.aarch64 : Disk Pool Manager (DPM) plugin for the Globus GridFTP server
erlang-inets.aarch64 : A set of services such as a Web server and a ftp client etc
erlang-ssh.aarch64 : Secure Shell application with sftp and ssh support
filezilla.aarch64 : FTP, FTPS and SFTP client
lftp.aarch64 : A sophisticated file transfer program
perl-Net-SFTP-Foreign.noarch : SSH File Transfer Protocol client
rssh.aarch64 : Restricted shell for use with OpenSSH, allowing only scp and/or sftp
wget.aarch64 : A utility for retrieving files using the HTTP or FTP protocols

```

Name and summary matches only, use "search all" for everything.

从列出的这些软件包里可以看到有一个软件包叫 vsftpd.aarch64,这个就是 Linux 系统上经常安装的 ftp 服务器。

(2) yum install: 安装软件包,用法如下:

```
yum install -y vsftpd
```

安装 vsftpd 软件包,因为使用了-y 参数,安装的时候不需要确认而直接安装。

(3) yum update: 更新软件包,用法如下:

```
yum update vsftpd
```

更新 vsftpd 软件包。

(4) yum remove: 删除软件包,用法如下:

```
yum remove vsftpd
```

删除 vsftpd 软件包,删除前需要确认。

(5) yum update: 更新所有软件包。

5. 压缩及解压缩命令

tar: 压缩及解压缩命令,根据其后的参数不同具有不同的功能,常用参数如下:

-z: 是否同时具有 gz 属性。

-x: 解压缩、提取打包的内容。

-c: 建立一个压缩、打包文档。

-v: 显示压缩或者打包的进程。

-f: 使用文件名,在 f 后面接压缩后的文件名字或者要解压的文件名字。

把 a.conf、b.conf、c.xml 文件打包到 tot.tar.gz 文件,使用 gzip 压缩,显示压缩的进程,命令如下:

```
tar -zcvf tot.tar.gz a.conf b.conf c.xml
```

解压缩文件 tot.tar.gz 到当前目录,命令如下:

```
tar -xvf tot.tar.gz
```

6. 服务相关命令

服务管理命令一般使用 systemctl,常用用法如下:

```
systemctl start vsftpd
```

启动服务 vsftpd。

```
systemctl status vsftpd
```

查看 vsftpd 服务状态,在回显里通过 Active 的值查看当前状态:

```
[root@ecs-kunpeng ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2020-12-01 09:14:42 CST; 3s ago
     Process: 3356 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
    Main PID: 3357 (vsftpd)
      CGroup: /system.slice/vsftpd.service
              └─3357 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Dec 01 09:14:42 ecs-kunpeng systemd[1]: Starting Vsftpd ftp daemon...
Dec 01 09:14:42 ecs-kunpeng systemd[1]: Started Vsftpd ftp daemon.
```

通过上面的回显可以看到 Active 的状态值为 active (running),表示处于运行状态。

停止 vsftpd 服务,命令如下:

```
systemctl stop vsftpd
```

重新启动 vsftpd 服务,命令如下:

```
systemctl restart vsftpd
```

启用开机自启动 vsftpd 服务,命令如下:

```
systemctl enable vsftpd
```

取消开机自启动 vsftpd 服务,命令如下:

```
systemctl disable vsftpd
```

7. 其他常用命令

(1) passwd: 修改密码命令。

修改当前用户密码,命令如下:

```
passwd
```

修改用户 tom 的密码,命令如下:

```
passwd tom
```

(2) date: 查看当前时间。

(3) scp: 不同主机之间复制文件。

把当前主机/opt/data/a.conf 文件复制到 192.168.1.1 对应的主机的/opt/data/目录下,在执行的时候,会要求输入 192.168.1.1 主机的 root 密码,命令如下:

```
scp /opt/data/a.conf root@192.168.1.1:/opt/data/
```

把 192.168.1.1 对应的主机的/opt/data/a.conf 文件复制到当前主机的/opt/data/目录下,在执行的时候,会要求输入 192.168.1.1 主机的 root 密码,命令如下:

```
scp root@192.168.1.1:/opt/data/a.conf /opt/data/
```

3.3 鲲鹏论坛

鲲鹏论坛是最重要的鲲鹏架构交流社区,任何与鲲鹏架构相关的问题都可以在上面讨论,华为有值班的专家关注论坛,可以第一时间回复求助的问题。鲲鹏论坛页面如图 3-4 所示。

在论坛发表帖子的时候,单击论坛右上角的“发表主题”按钮,在发表帖子页面可以选择帖子的主题,需要特别注意的是当有问题需要求助的时候,要将主题分类为“问题求助”,如图 3-5 所示。

然后按照帖子自动生成的问题描述步骤,详细写出问题,并附上截图、日志,最后提交即可。

华为专家看到求助帖后会第一时间进行回复,一般十几分钟就会有人回复,回复后的求助帖子会有专门的“专家已回复”标志,问题解决完毕的帖子会有“已结帖”标志,如图 3-6 所示。

除了日常的交流外,鲲鹏论坛还定期举办各种活动,例如直播、免费体验、优惠认证等,各种奖品及奖项层出不穷,如图 3-7 所示。



图 3-4 鲲鹏论坛



图 3-5 问题求助发帖



图 3-6 专家回复



图 3-7 论坛活动

3.4 鲲鹏云服务

云服务是鲲鹏架构最重要的应用场景之一，目前华为云上提供的基于鲲鹏架构的服务有上百种，这里重点介绍以下 3 种典型应用，分别是弹性云服务器(Elastic Cloud Server, ECS)、裸金属服务器(Bare Metal Server)及鲲鹏云手机(Cloud Phone, CPH)。

3.4.1 弹性云服务器 ECS

基于鲲鹏架构的弹性云服务器是开发者最常使用鲲鹏架构服务器的方式，它是由鲲鹏 CPU、内存、云硬盘等硬件及在其上安装的操作系统组成的计算组件。

用户可以在华为云上通过弹性云服务器购买鲲鹏云服务器，然后可以像使用本地服务器一样使用鲲鹏云服务器，弹性云服务器的网址为 <https://www.huaweicloud.com/product/ecs.html>。

弹性云服务器可以根据需要进行配置变更，例如 CPU 的核心数、内存的大小、云硬盘的容量、网络的带宽等，真正做到了按需配置、按需使用。

在性能和性价比方面，鲲鹏架构的 ECS 具有显著的优势，根据华为云官方的测算，相比其他架构，鲲鹏架构的 ECS 多核整型性能领先 15%，综合性价比提升 30% 以上。对于相似性能和配置的 x86 架构 ECS 和鲲鹏架构的 ECS 价格对比如图 3-8 所示。

产品	区域	计费模式	购买量	价格	操作
弹性云服务器 ECS 1	华北-北京四	包年	1年,1台	¥9,090.00 省: ¥1,818.00	删除
规格	鲲鹏计算 鲲鹏通用计算增强型 kc1.2xlarge.4 8核 32GB		1年,*1	¥7,800 省: ¥1,560	
系统盘	高IO 40GB		1年,*1	¥140 省: ¥28	
弹性公网IP	全动态BGP 独享 按带宽计费 5Mbit/s		1年,*1	¥1,150 省: ¥230	
镜像	CentOS CentOS 8.0 64bit with ARM		1年,*1	¥0	
弹性云服务器 ECS 2	华北-北京四	包年	1年,1台	¥11,366.60 省: ¥2,776.60	删除
规格	X86计算 通用计算增强型 c3ne.2xlarge.4 8核 32GB		1年,*1	¥10,076.6 省: ¥2,518.6	
系统盘	高IO 40GB		1年,*1	¥140 省: ¥28	
弹性公网IP	全动态BGP 独享 按带宽计费 5Mbit/s		1年,*1	¥1,150 省: ¥230	
镜像	CentOS CentOS 8.0 64bit		1年,*1	¥0	

图 3-8 ECS 对比

在选用鲲鹏架构弹性云服务器的时候,需要注意根据实际的业务情况进行有针对性选择,特别是 CPU 核心数和内存的比例及硬盘的类型,总体来说,鲲鹏架构 ECS 分为 4 个大类,分别是鲲鹏通用计算增强型、鲲鹏内存优化型、鲲鹏超高 I/O 型、鲲鹏 AI 推理加速型。

1. 鲲鹏通用计算增强型

该型号搭载鲲鹏 920 处理器及 25GE 智能高速网卡,配置比较均衡,CPU 最多支持 60 核心,内存最高支持 192GB,适合企业、政府、互联网等各种业务类型。

2. 鲲鹏内存优化型

该型号搭载鲲鹏 920 处理器及 25GE 智能高速网卡,和通用计算增强型相比,内存配置更高,CPU 最多支持 60 核心,内存最高支持 480GB,适合对内存要求比较高的各种业务。

3. 鲲鹏超高 I/O 型

该型号搭载高性能 NVMe SSD 本地磁盘,单盘 3.2TB,读吞吐量 2.9GB/s,写吞吐量 1.9GB/s,提供高存储 IOPS 及低读写时延,CPU 最多支持 64 核心,内存最高支持 228GB,适合高性能关系数据库、NoSQL 数据库及 Elasticsearch 搜索等业务场景。

4. 鲲鹏 AI 推理加速型

该型号配备 Atlas300 加速卡,该加速卡以华为昇腾 310(Ascend 310)芯片为核心,具有低功耗、高算力的特点,CPU 最多支持 48 核心,内存最高支持 96GB,加速核心支持最多 12 个 Ascend 310,适用于 AI 推理计算等业务场景。

鲲鹏架构 ECS 的具体购买步骤可以参考 4.1.4 节的内容。

3.4.2 裸金属服务器 BMS

裸金属服务器本质上是物理服务器,和 ECS 共享计算资源不同,BMS 独占计算资源,这就保证了极高的安全性,因为没有虚拟化的性能开销和特性损失,它可以发挥物理机器几乎全部的能力。

鲲鹏架构的裸金属服务器搭载两个鲲鹏 920 CPU,内存最高可达 1TB,为核心数据库、关键应用系统、高性能计算、大数据等业务提供卓越的计算性能及数据安全。

裸金属服务器网址为 <https://www.huaweicloud.com/product/bms.html>。

3.4.3 鲲鹏云手机

鲲鹏云手机本质上是一台包含原生安卓操作系统,具有虚拟手机功能的云服务器。具体实现是基于华为云裸金属服务器,在上面运行 EulerOS 作为 Host OS,在 Host OS 中运行 MonBox 生成容器,MonBox 类似 x86 架构下的安卓容器 AnBox,但是性能更强,是华为针对鲲鹏架构量身定制的。在容器中运行 AOSP 镜像,这样就虚拟出了一台云手机。鲲鹏架构的裸金属服务器也是基于 ARM 的,和手机系统一致,这样就没有指令转换的性能损失,一台裸金属服务器可以虚拟出多台鲲鹏云手机,华为云 HDP 规格显示,一台配备了两颗鲲鹏 916 处理器的裸金属服务器可以虚拟出 60 台云手机,而一台配备了两颗鲲鹏 920 处理器的裸金属服务器可以虚拟出高达 100 台云手机。

鲲鹏云手机运行在云端,具有强大的计算能力,同时集成了多张 GPU 显卡,可以提供专业的图形图像处理能力,适合云游戏、移动办公、App 仿真测试等多种业务场景。

鲲鹏云手机网址为 <https://www.huaweicloud.com/product/cloudphone.html>。

3.5 鲲鹏创新中心

为更好地建立鲲鹏软件生态,华为联合各省、市政府在当地合作建立鲲鹏创新中心,聚合区域产业合作伙伴,提供华为鲲鹏生态产品认证、应用迁移支持、行业示范、人才培养、标准孵化等服务。目前已经在全国建立了 18 个鲲鹏创新中心,详细信息如表 3-1 所示,当地的企业及鲲鹏开发人员可以向鲲鹏创新中心寻求支持。

表 3-1 鲲鹏创新中心

省份	鲲鹏创新中心名称	地 址
北京	北京鲲鹏联合创新中心	北京朝阳区
天津	天津鲲鹏生态创新中心	天津滨海高新区
山西	山西鲲鹏生态创新中心	山西智创城 NO.1 暨清控创新基地
陕西	陕西鲲鹏生态创新中心	西安高新区
河南	中原鲲鹏生态创新中心	郑东新区智慧岛
四川	四川鲲鹏生态创新中心	成都天府新区
重庆	鲲鹏计算产业生态重庆中心	重庆西永微电子产业园区
湖北	长江鲲鹏生态创新中心	武汉东湖新技术开发区未来科技城
江苏	江苏鲲鹏生态创新中心	南京市江北新区
上海	上海“鲲鹏+昇腾”生态创新中心	上海徐汇区 AI 大厦
浙江	浙江省鲲鹏生态创新中心	杭州市滨江东方通信园
浙江	宁波鲲鹏计算生态创新中心	宁波市软件产业园
福建	福建鲲鹏生态创新中心	福州滨海新城
湖南	湖南省鲲鹏生态创新中心	长沙湘江新区
贵州	贵州省鲲鹏生态创新中心	贵阳
广西	中国-东盟信息港鲲鹏生态创新中心	南宁五象新区
广东	广州“鲲鹏+昇腾”生态创新中心	广州市天河区
广东	鲲鹏产业源头创新中心	深圳湾科技生态园