

基于龙芯平台的 Docker 评测与分析^①



吴平凡^{1,2}, 刘显德¹, 吴少刚²

¹(东北石油大学 计算机与信息技术学院, 大庆 163318)

²(江苏航天龙梦信息技术有限公司, 苏州 215500)

通讯作者: 吴平凡, E-mail: moxie96@126.com

摘要: 对轻量级技术的 Docker 技术进行了概述, 针对 Docker 1.13+ 版本基于龙芯平台进行了移植并集成到 Fedora 28 系统中, 制作测试镜像对新版本 Docker 方案进行了性能测试分析, 剖析了不同容器数量下的性能变化趋势和容器的性能瓶颈. 通过龙芯单路、双路、四路服务器和 AMD Ryzen 5 (2400 GB) 主机进行了容器内的性能对比测试实验, 肯定了龙芯平台上的新版 Docker 方案的稳定性, 并分析了龙芯 3A3000 芯片与 Ryzen 5 (2400 GB) 芯片相近主频下的性能差异, 展望了国产 CPU 芯片事业的发展前景.

关键词: 龙芯平台; Docker; 镜像制作; Fedora; 性能分析

引用格式: 吴平凡, 刘显德, 吴少刚. 基于龙芯平台的 Docker 评测与分析. 计算机系统应用, 2020, 29(4): 58-64. <http://www.c-s-a.org.cn/1003-3254/7346.html>

Evaluation and Analysis of Docker Based on Loongson

WU Ping-Fan^{1,2}, LIU Xian-De¹, WU Shao-Gang²

¹(School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

²(Jiangsu Lemote Technology Co. Ltd., Suzhou 215500, China)

Abstract: The technology of Docker based on the lightweight technology was summarized. The Version 1.13+ of Docker was transplanted and integrated into Fedora 28 system based on Loongson. The test image was made to analyze the performance of the new version of Docker scheme, the trend of performance and the bottlenecks of container's function under different number of containers were analyzed. By comparing the performance in containers of Loongson single-way, double-way, four-way servers and a machine with AMD Ryzen 5 (2400 GB), the stability of the new version of Docker scheme on Loongson was affirmed. The difference of performance between Loongson 3A3000 and Ryzen 5 (2400 GB) under the near basic frequency was analyzed. The prospect of industry about domestic CPU chip is expected.

Key words: Loongson platform; Docker; images making; Fedora; performance analysis

云计算中最关键的技术就是虚拟化技术, 用以实现不同独立应用间的资源共享. 传统的虚拟机技术由于过度的抽象化在实际生产环境中部署繁琐, 资源消耗过多, 启动速度慢. 自 2013 年以来, 随着以 Docker 为代表的新一代容器技术的快速发展, 已成为各大云计算厂商和云计算开发者的首选. 对比虚拟机技术,

Docker 具有应用部署灵活, 资源消耗少, 启动速度快等优点, 凭借轻量级的特性在虚拟化领域形成了颠覆性的影响^[1]. 近年来, 国内大量互联网公司也纷纷进行以新一代容器技术为核心进行技术转型, 阿里在 2017 年 11 月正式开源了其自研容器 Pouch Container. 而在 2018 年 5 月的全球软件与运维技术峰会上, 知乎计算

① 基金项目: 江苏省科技成果转化专项 (BA2018074)

Foundation item: Special Fund for Transformation of Scientific and Technological Achievements of Jiangsu Province (BA2018074)

收稿时间: 2019-08-28; 修改时间: 2019-09-29, 2019-10-11; 采用时间: 2019-10-21; csa 在线出版时间: 2020-04-05

平台宣布利用 Docker 已实现全部业务和服务的容器化运行. 国内外的 Docker 相关研究^[2,3]更是层出不穷, 这些现象表明新一代容器技术火热的发展趋势.

在龙芯平台虚拟化技术研究方面, 早在 2013 年蔡万伟等人^[4,5]就已经做了大量关于 MIPS 架构下的内存虚拟化研究. 2016 年王篁^[6]为系统级的二进制代码兼容性优化提供了解决方案. 2019 年 4 月龙芯公司宣布龙芯 KVM 虚拟机产品正式发布. 而关于容器技术方面, 龙芯平台上 Docker 方案还是停留在比较老旧的 Fedora21 系统的 Docker 老版本上, 而新版 Docker 需要进一步在龙芯平台上得以支持.

鉴于 X86 的 Fedora28 系统默认安装源为 1.13.1 版本的 Docker 方案, 所以本文针对该版本进行了移植并集成到龙芯平台的 Fedora28 系统中, 并测试分析了新版 Docker 方案在龙芯平台下的性能表现, 凭借容器内测试程序稳定性更高的优势, 比较了容器内同主频下龙芯 3A3000 与 AMD Ryzen 5 (2400 GB) 的性能表现.

1 Docker 技术概述

作为一个热门技术, Docker 正处在快速发展的阶段, 新版本和新特性不断涌现^[7]. 1.13 版本作为 Docker 分离社区版和企业版前的最后一个版本, Docker 方案的整体架构已经逐渐稳定下来, 如图 1 所示.

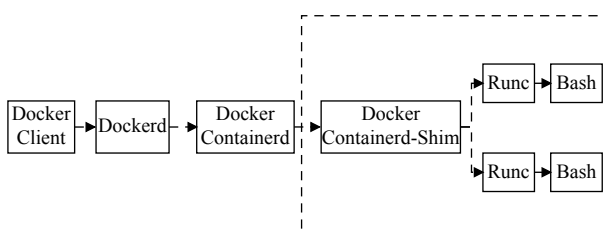


图 1 容器运行时架构图

如今的 Docker^[8]方案, 对于容器整个生命周期的管理操作都由 Containerd 模块^[9]完成, Dockerd 将创建容器和运行容器等操作直接发送给与 Containerd 来完成, Containerd 收到请求后将启动一个 Containerd-Shim 进程来管理一个运行的容器, 通过 Runc 模块与容器进行交互. Containerd 和 Runc 结合起来成为了容器服务标准化的两个重要部分. 同时用户可以直接在 Containerd 和 Runc 上建立包括 Microsoft Azure、VMWare 在内的多种上层应用, 并不仅仅局限于 Docker 容器平台. 其中的 Runc 模块是从 Docker 的 Libcontainer 中演化过来的, 相当于 Libcontainer 模块结合上一个小型的客户端用以实现容器的启动停止和资源隔离等功能. 作为应用程序级的虚拟化技术, 如今的 Docker 中通过 Runc 模块, 即使绕过 Daemon 守护进程依然能够直接运行容器, 某种程度上来说已经不依赖于 Docker 本身了^[10].

2 Docker 方案评测

龙芯平台上的 Fedora28 系统已经发布, 故本次测试将在移植 Docker 1.13+版本到 28 系统上后, 使用测试工具对 Docker 方案进行测试, 其中涉及的机器配置如表 1.

2.1 Docker 移植

在进行具体的测评工作之前需要对相应的 Docker 方案进行 MIPS 平台的编译工作. 由于官方提供的安装方法是在容器内自动安装, 过程需要从 Docker 官方镜像仓库中拉取需要的 Image 镜像, 涉及镜像数量较多, 且大量镜像需要在 MIPS 平台上重新制作, 故本次实验选择手动编译的方式进行移植, 根据龙芯平台 Fedora28 系统的差异性对 Docker 源码进行调整, 添加 MIPS 架构入口标识, 优化不同架构平台的参数设置.

表 1 龙芯机器硬件参数表

类型	规格		
名称	龙芯单路主机	龙芯双路服务器	龙芯四路服务器
处理器	Loongson-3A3000 (4 cores)	两片 3A3000 (8 cores)	四片 3A3000 (16 cores)
主频	1.5 GHz	1.45 GHz	1.45 GHz
内存	4 x DDR3 DIMM (8 GB)	8 x DDR3 DIMM (16 GB)	16 x DDR3 DIMM (32 GB)
桥片	AMD RS780E+SB710	AMD SR5690+SP5100	AMD SR5690+SP5100
OS	Fedora 28	Fedora 28	Fedora 28
Kernel	4.19.5-2.fc28.lmote.1.mips64el	4.19.5-2.fc28.lmote.1.mips64el	4.19.5-2.fc28.lmote.1.mips64el

Docker 方案编译工作完成后, 测试准备工作仍未完成, 由于平台架构的差异性, 测试工作中涉及的所有

镜像需要重新设计. 尤其是新版本 Fedora28 系统的基础镜像, 为预防可能出现的镜像冗余问题, 需要重新制

作一个单独的精简系统,并保证新系统安装源的维护.制作的 Fedora28-base 镜像已经推送到 Docker 官方仓库上,并同步更新下载源.

2.2 测试镜像设计

为了对搭建好的 Docker 平台进行综合性能评估,分析龙芯 CPU 上的 Docker 性能表现,在综合考量大量性能测试工具后,基于实验可重复性和简洁性的考虑,选择结合 Unixbench 工具制作测试镜像.因为 Unixbench 工具是一款非常经典的系统基本性能测试工具,相对其他一些测试工具而言,它的测试项更为全面,而且能自动化执行一系列的测试命令,这更加契合测试镜像的制作初衷,提高了测试效率. Unixbench 包括系统调用、IO 读写、进程、2D、3D、管道、运算、C 库等在内的多种性能测试,测试结果不仅仅取决于 CPU、内存、硬盘,还关乎硬件、开发库和编译器等,能较为全面地综合评价系统各方面性能.而且 Unixbench 能将测试后的数据计算成相应的分数以供用户参考,使测试结果更加直观简洁.

重要的是, Unixbench 支持多 CPU 系统的测试,默认的测试行为是两次,第一次进行单进程测试 (Running 1 Parallel Copy of Tests),第二次则是进行当前系统 CPU 个数的测试 (Running N Parallel Copy of Tests). 这样的测试是为了测试系统多任务下并行处理的性能,以更为客观地表现出实际数据结果. 主要测试项如下:

(1) Dhystone 测试: 该测试用于衡量和比较计算机 CPU 的性能,主要侧重于整数计算,没有浮点计算,受到编译器和链接器选项,软硬件设计等影响较大.

(2) Whetstone 测试: 这项主要用于测试浮点数的运算和效率,包含大量 C 语言函数、浮点数的运算操作和若干科学计算的经典性能模块等.

(3) Execl Throughput 测试: Execl 吞吐量测试,用于测试 Execl 函数每秒调用次数, Execl 函数是 Exec 函数家族的一部分,该函数族提供了在一个进程中启动另外一个执行程序的方法.

(4) Pipe Throughput 测试: 用于测试进程间的通信,即一秒钟一个进程写 512 bite 到一个管道中并读回来的次数.

(5) Pipe-based Context Switching 测试: 基于管道的上下文交互测试,这是两个进程通过管道进行通信的次数.

(6) Process Creation 测试: 用于测试每秒创建进程

并立刻收回的次数.

(7) Shell Scripts 测试: 用于测试一定时间内进程启停 Shell 脚本的次数.

(8) System Call Overhead 测试: 用于测试系统调用的性能,即重复执行 getpid() 函数调用时进入和离开内核的时间.

2.3 不同容器数量的测试结果

测试分别在单核系统环境和四核系统环境下进行以便分析容器下虚拟化性能损耗. 首先对龙芯单路主机进行 Unixbench 测试作为基准值记为 Container Number = 0, 然后在不同数量的 Docker 容器内同时运行测试镜像进行性能测试.

这次实验中,设容器编号 (Container Number) 为 0 时在宿主机上运行测试,容器编号为 N 时在同时运行的 N 个容器中运行测试,对于每次测试重复执行 3 次以降低误差的影响,每一项测试结果取所有的数据结果的平均值为最终值,即每次最终的测试数据 $Data_N$ 计算如下:

$$Data_N = \frac{1}{3} \sum_{i=1}^3 \frac{1}{N} \sum_{j=1}^N u_{ij} (N \geq 1) \quad (1)$$

式中, u_{ij} 表示在 N 个容器同时运行情况下,第 j 个容器第 i 次的测试数据.

实际测试数据表如表 2.

由于 Unixbench 工具出于测试系统单任务性能与多任务下并行处理性能的目的,测试行为中有两项测试,即单进程测试 (Running 1 Parallel Copy of Tests) 和当前系统 CPU 个数的多进程测试 (Running N Parallel Copy of Tests, [CPU(N)]). 单进程测试中所有的处理器只处理一个测试进程,而多进程测试中所有的处理器同时运行处理器数量个的测试进程. 由于单核系统环境环境下无需进行并行处理的测试,故只有一个结果,这个测试结果既可以看作是单核系统下的单测试任务结果,也可以看作是多测试任务的结果. 以下的数据分析中以多测试任务结果为主.

2.4 测试结果分析

Dhystone 测试是测试处理器运算能力的基准程序之一,是各个测试项中最能反映 CPU 能力的测试项,所以着重分析这一项测试的结果,来研究虚拟化在龙芯上的影响.

表2 不同容器数量下的性能测试结果

System Benchmarks Index Values	4 CPU in system; running 4 parallel copy of tests					1 CPU in system; running 1 parallel copy of tests					4 CPU in system; running 1 parallel copy of test				
	CPU(4) SCORE					CPU(1) SCORE					CPU(4) SCORE[for one test]				
Containers Numbers	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
Dhrystone 2 using register variables	2857.2	2971.1	1631.5	1271.9	753.5	709.9	740.0	404.5	326.8	244.5	718.4	745.0	744.6	740.3	742.4
Double-Precision Whetstone	717.1	801.0	798.3	797.2	798.6	180.0	199.9	199.4	199.9	199.9	179.8	199.2	201.2	201.6	199.2
Execl Throughput	1142.8	1165.9	713.7	563.7	405.5	359.9	380.8	194.5	148.1	112.2	339.1	353.2	353.9	350.6	381.9
Pipe Throughput	1509.9	1512.0	824.2	601.5	452.4	368.0	364.9	195.9	140.9	103.9	378.8	378.0	380.4	380.0	377.6
Pipe-based Context Switching	479.2	464.5	434.5	344.8	250.7	248.0	240.0	129.0	92.5	67.9	112.8	108.4	113.5	100.5	115.2
Process Creation	817.5	776.2	525.8	381.9	290.0	307.4	266.1	133.3	96.6	72.8	321.4	309.0	294.1	221.8	272.5
Shell Scripts (1 concurrent)	1691.0	1673.8	871.5	586.4	427.2	473.5	462.9	225.5	150.5	109.7	659.5	648.4	567.3	516.1	421.3
Shell Scripts (8 concurrent)	1653.4	1643.8	816.2	578.5	428.9	442.1	431.9	206.3	149.2	109.8	1466.8	1459.7	835.2	571.9	400.3
System Call Overhead	1216.3	1216.5	679.3	508.5	384.4	353.4	353.7	187.6	135.6	98.3	346.0	347.0	345.8	346.2	345.4
System Benchmarks Index Score	975.2	961.3	792.0	642.6	509.1	400.8	444.0	240.8	185.7	142.5	436.7	439.1	463.2	427.2	418.4

从表2中,以测试处理器运算能力的基准程序Dhrystone测试的结果可以看出,四核环境下的测试中,随着同时运行测试程序的容器数量增加,机器的字符串处理性能几近线性下降。如图2。而在单核测试中,多容器下的性能以接近 $y=1/x$ 函数的形式呈曲线下降,如图3,与基准值(单容器测试数据)相比,双容器下Dhrystone测试性能约为54.67%,三容器下Dhrystone测试性能约为44.16%,四容器下Dhrystone测试性能约为33.04%。而其他的大部分测试项,无论在四核还是单核条件下都有相似的结果,性能并不是呈线性下降的,比如Execl函数测试和管道吞吐量测试等,因为资源控制机制发挥了很大影响,它对CPU、内存、磁盘等计算资源的使用进行了调控。

而从表2的第3部分可以看到,在四核环境下的单进程测试(Running 1 Parallel Copy of Tests)中,Dhrystone测试的结果并没有变化,这是由于四核环境下的系统运算能力并未完全发挥出来,即使是在4个容器内同时运行测试任务,也只是最大限度地体现了四核下的运算能力,Dhrystone测试的单进程测试平均结果为473.0,稍高于单核测试中的双容器测试结果14.7%,多进程测试平均结果为492.2,高于基准值16.6%。

而Whetstone测试结果来看,数据结果十分稳定,即使在八容器的测试实验中,Whetstone测试在单进程测试中的平均结果也是200.2,多进程下结果是794.5,误差几乎可以忽略不计,再结合Dhrystone测试的结

果,可以确定容器虚拟化给CPU的计算能力带来损耗非常小。

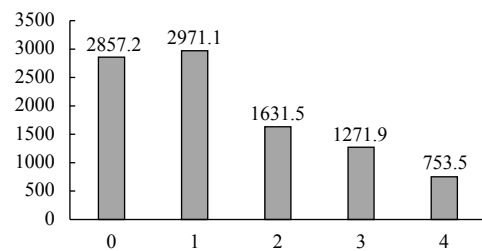


图2 Dhrystone 测试 CPU(4) for 4 tests

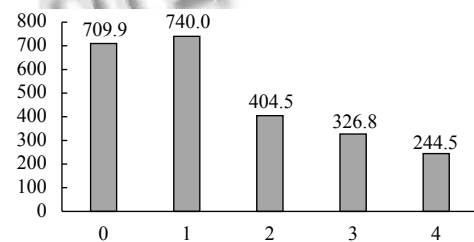


图3 Dhrystone 测试 CPU(1) for 1 test

就龙芯单路主机上的测试结果和单容器测试的基准值比较,本文发现,在四核环境下,主机上的Dhrystone测试结果是基准值的96.17%,单核环境下是基准值的96.0%,而其他的一些测试项并未有这样的结果,因此推测,这是由于容器内的镜像系统是轻量化的精简系统,在测试程序执行过程中CPU对于其他可能存在的负载程序的支持更为轻松,所以测试结果显示的相关计算能力会较为优越,这一结果也更加体现了Docker

镜像的制作对于实际应用中可能产生重要影响,一个优越的轻量化镜像能更好地提升容器集群的性能和实际运行的流畅度。

3 龙芯平台与 X86 平台下的测试比较

3.1 与锐龙 5 处理器的对比测试

作为比较对象, X86 平台下选择的是搭载 AMD 锐龙 5 处理器的机器, 硬件参数表如表 3。

容器内的测试更不容易受到误差因素干扰, 可靠性和稳定性更强, 因此在实际实验中, 分别在龙芯的单路服务器、双路服务器和四路服务器上的测试容器内

进行 Unixbench 测试, 并在搭载 AMD Ryzen 5 2400 GB 芯片的 X86 机器上运行同样的测试容器, 由于 Unixbench 工具的测试特性分为两项测试结果, 相关测试结果如表 4 和表 5 所示。

表 3 X86 平台硬件参数表

类型	规格
名称	AMD 锐龙
处理器	AMD Ryzen 5 2400 GB
主频	3.6 GHz(默认)
内存	16 GB
OS	Fedora 28
Kernel	5.0.5-100.fc28.x86_64

表 4 Unixbench 测试结果 (Running 1 Parallel Copy of Tests)

	MIPS_CPU(4)	MIPS_CPU(8)	MIPS_CPU(16)	X86_CPU(8)
Dhrystone 2 using register variables	712.9	687.9	692.4	1795.6
Double-Precision Whetstone	216.5	207.9	192.4	556.2
Execl Throughput	365.9	214.9	173.8	523.5
Pipe Throughput	382.1	368.8	393.1	687.8
Pipe-based Context Switching	117.6	79.8	57.7	211.4
Process Creation	298.2	191.5	160.6	543.7
Shell Scripts (1 concurrent)	639.5	432.7	448.6	946.4
Shell Scripts (8 concurrent)	1393.7	1065.4	936.3	2831.4
System Call Overhead	335.4	323.5	361.4	988.5
System Benchmarks Index Score	452.6	364.6	335.2	913.7

表 5 Unixbench 测试结果 (Running N Parallel Copy of Tests [CPU(N)])

	MIPS_CPU(4)	MIPS_CPU(8)	MIPS_CPU(16)	X86_CPU(8)
Dhrystone 2 using register variables	2929.2	5408.5	10 624.0	8501.4
Double-Precision Whetstone	908.1	1650.1	2793.0	3681.4
Execl Throughput	1269.1	934.0	671.3	2375.1
Pipe Throughput	1575.8	2931.3	6012.0	3840.0
Pipe-based Context Switching	1175.8	980.9	1001.3	1400.0
Process Creation	772.3	715.5	700.2	2320.0
Shell Scripts (1 concurrent)	1567.2	1731.5	1981.4	3558.0
Shell Scripts (8 concurrent)	1654.4	1790.8	2021.3	3737.0
System Call Overhead	1221.4	702.6	439.1	4726.3
System Benchmarks Index Values	1104.2	1108.1	1091.8	2815.5

由于 Ryzen 5(2400 GB) 处理器上的 CPU 主频是动态的, 因此为保证测试数据的客观性和对比实验的可行性, 在 BIOS 选项中设置了搭载 Ryzen 5 处理器的主机主频, 然后通过不断读取 /proc/cpuinfo 文件信息, 确认主机的主频均值在 1.45 GHz 左右, 与龙芯双路服务器主频相近。

龙芯双路、四路服务器下 CPU 主频为 1.45 GHz, 单路主机为 1.5 GHz, 实际测试数据表如表 4 和表 5。

3.2 测试结果分析

在单进程测试下 (Running 1 Parallel Copy of

Tests), 由于只有一个测试进程, 所以无论处理器的核心数量是多少, 都只有一个测试进程在执行, 相当于在一个核心上进行性能测试, 故首先可以从最能体现芯片性能的两项测试 Dhrystone 测试和 Whetstone 测试开始分析, 如图 4 所示。

首先从单个芯片的性能数据开始比较, 主要以 MIPS_CPU(8) 和 X86_CPU(8) 下的结果进行比较。根据表 4 和图 4, 可以看到在 Dhrystone 测试方面, 双路服务器中单核心龙芯性能分数为 687.9, X86 下的测试分数为 1795.6, 龙芯在容器内进行 Dhrystone 测试的执

行效率大约是 Ryzen 5 (2400 GB) 的 38.3%, 而在与科学计算性能相关的 Double-Precision Whetstone 测试中, 龙芯的运算效率大约是 Ryzen 5 (2400 GB) 的 37.4% 左右, 由两个测试项数据结合确定了龙芯双路服务器在运行单个测试进程时 CPU 计算性能约为搭载 Ryzen 5 主机的 40% 左右. 但这仅仅是单个测试进程的结果, 而基于 Unixbench 测试工具的设计理念, 多进程测试的结果更为贴近实际的应用场景表现出来的性能, 因此需要着重分析多进程测试的数据结果. 根据表 5, 可以得到多进程测试下的 Dhrystone 测试和 Whetstone 测试数据结果, 如图 5.

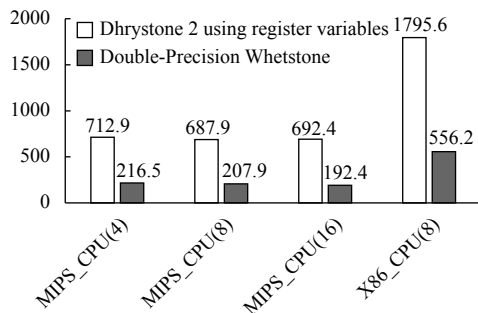


图 4 Dhrystone 测试和 Whetstone 测试数据 (1 parallel test)

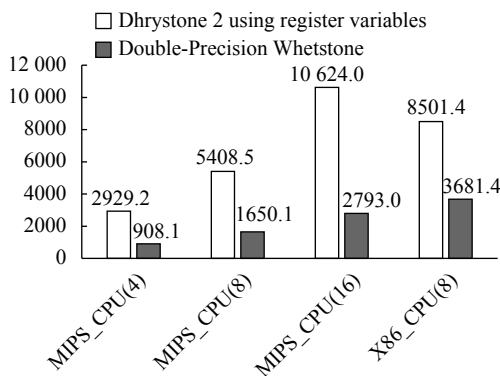


图 5 Dhrystone 测试和 Whetstone 测试数据 (N parallel tests)

从图 5 中很清晰地看到, 在龙芯双路服务器中, Dhrystone 测试的整数计算性能约为搭载 Ryzen 5 主机的 63.6%, Whetstone 测试的浮点数计算性能约为 44.8%, 后续在进行验证性测试时发现, 数据结果基本稳定在测试数据的附近区间内, 误差大约在 3% 左右浮动, 说明数据是可靠的. 因此可以确定, 处理器核心数量越多, 基于 Dhrystone 测试和 Whetstone 测试得到的性能分数越高, 机器整数运算性能和浮点数运算能力越强, 而相等的处理器核心数量下, 龙芯 3A3000 芯

片的相关计算性能约为 Ryzen 5(2400 GB) 芯片的 40~60%. 其中多进程测试下, 与 Ryzen 5 主机相比, 龙芯双路服务器的性能表现从 38.3% 提高到了 63.6%, 这是由于龙芯双路服务器是八块 CPU 芯片, 而 Ryzen 5 主机只是四块 CPU 八核心的处理器结构. 其他项的测试可见图 6 和图 7.

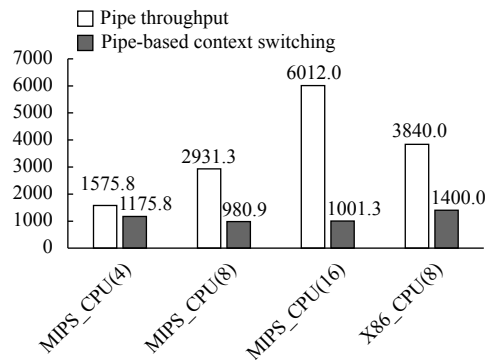


图 6 Pipe 相关测试数据 (N parallel tests)

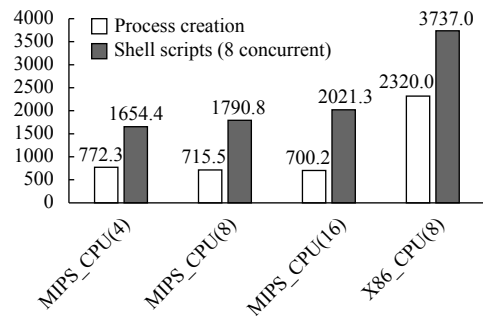


图 7 进程相关测试数据 (N parallel tests)

管道吞吐 (Pipe Throughput) 和交互测试 (Pipe-based Context Switching) 两项测试中, 龙芯双路服务器上的性能分数分别是 2931.3 和 980.9, 大约为 Ryzen 5 (2400 GB) 下的 76.3% 和 70.1%. 这两项测试表明在相近主频下, 龙芯双路服务器在进程通信效率上大约为 Ryzen 5(2400 GB) 主机的 70%~76%. 其他与系统性能相关测试的还有 Process Creation 测试, 用于测试每秒创建进程并回收的次数, 龙芯双路服务器性能约为 Ryzen 5 (2400 GB) 的 30% 左右; Shell Scripts 测试用于测试一定时间内进程可启停 Shell 脚本的次数, 龙芯双路服务器性能约为 Ryzen 5 (2400 GB) 的 48% 左右. 因此综合来看, 同样的 Fedora28 系统的容器运行环境下, 龙芯双路服务器的系统综合性能大约为 Ryzen 5 (2400 GB) 的 50% 左右.

3.3 Docker 方案综合分析

首先, Docker 方案在移植到龙芯 Fedora28 系统上之后, 运行状态良好, 稳定性可靠性大幅提升, 同时经过大量验证性实验之后可以肯定, 容器内的性能分数稍高于主机环境约 5% 左右, 在排除误差带来的影响之后, 体现了容器本身的轻量级技术的特性, 同时也印证了镜像构建的重要性, 可以预测一个优越的轻量化镜像能更好地提升容器集群的性能和实际运行的流畅度, 所以在后续构建和维护龙芯平台的 Docker 镜像仓库中同样需要秉持认真负责的态度。

其次, 本文发现多容器状态的测试数据并不稳定, 而且产生了一些性能损失, 这是因为容器对于资源管理功能的实现仍然深深依赖与 Linux 系统内核的自身特性, 由于 Docker 方案本身就是基于传统 LXC 技术实现的应用程序容器技术^[10], 旨在提供标准运行环境, 实现底层操作系统与物理主机的解耦. 因此, 在容器数量增加到一定程度后, 可以预见到容器间资源的不合理行为会变得十分频繁从而影响整个的容器集群. 从本次测试的多容器数量的性能趋势中可窥一斑, 所以在之后的实验研究中, 将把研究重点放到容器的动态调度算法和集群负载均衡策略优化上去, 通过容器调度策略等设计优化容器的实际应用环境。

同时, 根据借助容器环境在 Ryzen 5 (2400 G) 处理器上实现的对比实验, 本文发现, 目前主流的龙芯芯片 3A3000 与 Ryzen 5 (2400 GB) 芯片相比 CPU 计算性能约在 60% 左右, 这是龙芯双路服务器与八核的 Ryzen 5 (2400 GB) 机器在进行多进程测试时比较得到的结果. 如果就单进程测试而言, 龙芯 3A3000 的性能大约是 Ryzen 5 (2400 GB) 的一半, 如果综合考虑所有系统性能测试项, 龙芯双路服务器的综合评分大约有搭载 Ryzen 5 (2400 GB) 主机的 50% 左右。

4 结束语

经过分析本文得到以下结论: 一是龙芯平台上的新版 Docker 方案适应性良好, 稳定性可靠性大大提高, 并且发现轻量化的镜像会略微提升容器内隔离进程的运行效率, 这体现了容器本身的轻量级技术的特性, 同时也印证了镜像构建工作的重要性; 二是通过性能数据的渐变趋势, 可以预测高容器数量级下整个集群系

统性能会产生很大损失, 仅仅依赖 Linux 内核特性进行资源管理是远远不够的, 所以下一步研究将集中在容器集群模式下的负载均衡策略和动态调度算法设计上去; 三是在容器中, 经过与 Ryzen 5 (2400 GB) 芯片的对比实验发现, 根据 Dhrystone 测试和 Whetstone 测试结果, 龙芯 3A3000 芯片在 CPU 计算性能方面大约是 Ryzen 5 (2400 GB) 芯片性能的 40%~60% 左右, 多进程测试中 Dhrystone 测试性能约为 Ryzen 5 (2400 GB) 的 63.6%. 综合考虑所有系统性能测试项, 龙芯双路服务器的综合性能评分大约为搭载 Ryzen 5 (2400 GB) 主机的 50% 左右. 但是微架构优化后的下一代龙芯 3A4000 芯片即将推出, 性能方面将会有更大的提升, 国产化 CPU 事业会发展的越来越好。

参考文献

- 1 武志学. 云计算虚拟化技术的发展与趋势. 计算机应用, 2017, 37(4): 915-923. [doi: 10.11772/j.issn.1001-9081.2017.04.0915]
- 2 Martin A, Raponi S, Combe T, *et al.* Docker ecosystem-vulnerability analysis. Computer Communications, 2018, 122: 30-43. [doi: 10.1016/j.comcom.2018.03.011]
- 3 Di Tommaso P, Palumbo E, Chatzou M, *et al.* The impact of Docker containers on the performance of genomic pipelines. PeerJ, 2015, 3(3): e1273.
- 4 蔡万伟, 台运方, 刘奇, 等. 基于 MIPS 架构的内存虚拟化研究. 计算机研究与发展, 2013, 50(10): 2247-2252. [doi: 10.7544/issn1000-1239.2013.20111494]
- 5 蔡万伟, 台运方, 刘奇, 等. 基于 MIPS 架构的异构内存虚拟化方法研究. 高技术通讯, 2013, 23(9): 908-913. [doi: 10.3772/j.issn.1002-0470.2013.09.005]
- 6 王篁. 基于龙芯平台的虚拟机研究[博士学位论文]. 合肥: 中国科学技术大学, 2016.
- 7 Boettiger C. An introduction to Docker for reproducible research. ACM SIGOPS Operating Systems Review 49.1, 2015: 71-79.
- 8 Merkel D. Docker: lightweight linux containers for consistent development and deployment. Linux journal 2014.239, 2014: 2.
- 9 Anderson C. Docker [software engineering]. IEEE Software 32.3, 2015: 102-c3.
- 10 Combe T, Antony M, and Roberto DP. To docker or not to docker: A security perspective. IEEE Cloud Computing 3.5, 2016: 54-62.