气象大数据云平台仿真环境容器调度性能优化研究

吴鹏[1] 韩同欣[1*] 陈士旺[1] 聂元丁[1] 郑晓志[2]

(1 国家气象信息中心,北京 100081; 2 广州市海珠区气象局,广州 510240)

摘要 为实现2025年气象关键核心技术自主可控的目标,气象大数据云平台(简称天擎)建立了基于海光X86服务器和麒麟操作系统的仿真环境。在仿真平台运行中发现,基于容器技术的产品加工与流水线子系统容器调度性能较差,不能满足用户融入算法的时效要求。针对此问题,本文采用对比分析法,选取天擎仿真环境和业务环境的3种CPU芯片服务器和3种操作系统为研究对象,设计了一系列组合对比测试用例,找到了影响容器调度性能的关键因素—操作系统内核,并进一步分析了操作系统内核设置对系统实时性和吞吐量的影响以及适用的业务场景。最后给出了麒麟操作系统内核调整方法,通过调整内核设置,容器调度性能大幅提高,满足了产品加工系统的时效要求,为实现天擎的关键核心技术自主可控奠定基础。

关键词 气象大数据云平台; 自主可控; 麒麟操作系统; 容器调度

中图分类号: P409 **DOI**: 10.19517/j.1671-6345.20230182 **文献标识码**: A

引言

国家"十四五"规划和2035年远景目标纲要中提出要坚持创新在我国现代化建设全局中的核心地位,把科技自立自强作为国家发展的战略支撑[1]。气象"十四五"发展规划明确指出:要坚持创新驱动发展,加快气象科技自立自强,到2025年要实现关键核心技术自主可控。气象大数据云平台(以下简称"天擎")是利用云计算、大数据等技术构建的集气象数据收集分发、加工处理和存储管理及服务为一体的核心信息系统[2-4],支撑了气象观测、预报和服务等业务。为响应国家战略要求,实现气象"十四五"时期业务系统自主可控的目标,2022年天擎基于华为ARM芯片服务器,海光X86芯片服务器和麒麟操作系统建立了仿真支撑环境,开展基于国产基础软硬件的业务迁移适配、仿真开发和测试工作。

随着云计算技术的发展,云平台在教育、医疗、建筑、传媒等各领域得到了广泛的应用[5-11]。容器技术作为一种轻量级虚拟化技术,在云计算PAAS(Platform as a Service)平台中的部署和应用研究已经很多,众多学者基于容器技术提出了企业级私有云的实现方案[12-15]。资源调度是云计算的一个主要研究方向,目前已有很多对云计算资源和容器调度算法的研究[16-22]。例如,林伟伟和齐德昱[16]讨论了以降低云计算数据中心能耗为目标的资源调度方法,以提高系统资源利用率为目标的资源管理方法。邓广宏和张棋恒[21]提出了分层调度模型解决了实时容器和非实时容器混用情况下调度机制对实时容器造成的实时性降低问题。云计算和容器安全是云计算的另一个研究热点。例如,张玉清等[23]介绍了云虚拟化安全,云数据安全以及云应用安全的相关研究进展,指出了云计算环境安全未来发展趋势和后续研究方向。冯登国等[24]分析了云计算对信息安全领域中技术、标准、监管等各方面带来的挑战,

国家气象信息中心信息网络安全与"信创"技术研发创新团队(NMIC-202011-05)攻关任务与广东省气象局科学技术研究项目(GRMC2022Z05, GRMC2021XQ03)项目资助

作者简介:吴鹏,男,1982年生,硕士,高级工程师,主要从事气象通信、基础平台方面研究,Email:wupeng@cma.gov.cn

收稿日期:2023年6月2日;定稿日期:2024年1月12日

*通信作者,Email:hantx@cma.gov.cn

©《气象科技》编辑部(CC BY-NC-ND 4.0)

提出云计算安全参考框架及该框架下的主要研究内容。但目前容器等云计算技术在国产环境的部署和性能优化等方面的研究较少,按照国家相关要求,海光 X86、ARM 等芯片服务器和麒麟、统信等操作系统将逐渐成为未来国内企业生产环境的中坚力量,探讨云计算技术在国产平台的部署和优化,有非常重要的现实意义。

1　天擎仿真环境性能问题

天擎包含了数据交换与质控处理子系统,存储管理与服务子系统,产品加工与流水线子系统(以下简称"产品加工系统")等核心业务系统。天擎仿真环境的物理服务器使用了 2 种技术架构的 CPU 芯片—海光 X86 和华为鲲鹏 ARM,上面统一搭载麒麟 v10(以下称"Kylin")操作系统,其中的产品加工系统在操作系统之上部署了博云容器云环境,以支撑各类气象数据加工处理算法的实时调度和运行,如图 1 所示。

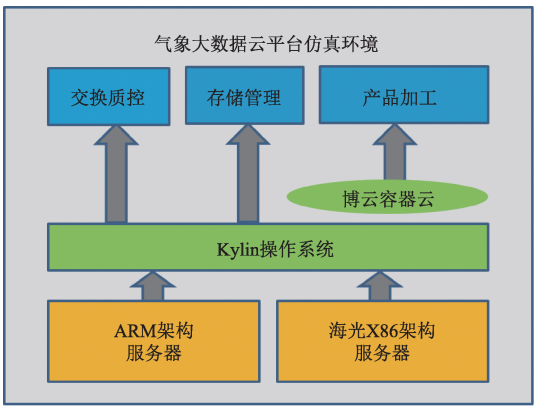

图 1　气象大数据云平台仿真环境系统平台

Fig. 1　System platform for simulation environment of meteorological big data cloud platform

天擎产品加工系统,实现了对各类气象数据加工处理算法的统一管理,并基于观测、预报和服务业务间的上下游关系建立了加工流水线,实现了加工处理任务的自动编排和统一调度。该系统基于消息总线和分布式调度算法,全面采用容器技术。相比虚拟机,容器具有轻量化,更加敏捷等优势[25],可以更好的支撑任务编排,资源负载均衡和弹性调度等业务需求。该系统的基本工作流程为:用户首先在系统注册算法,指定使用的数据源,所需数据到达后基于数据源驱动的容器自动创建,执行用户算法,生成相关产品。

根据业务需要,产品加工系统容器调度性能主要关注以下两个指标:①并发数:根据同一时刻需要执行的用户算法最大数量确定,现有系统已融入用户算法对容器数量需求最大值在 50 以内。同时考虑未来 3 年业务增长并发数需要达到 100。②调度时间:根据现有产品加工算法的时效要求确定。天擎产品加工系统容器并发调度具体性能需求见表 1。

表 1　产品加工系统容器调度性能需求

Table 1　Performance requirements for docker scheduling in product processing system

业务场景	平均调度时间/s	最大调度时间/s
50 并发,单节点	<2	≈2
100 并发,双节点	<2	≈2

经测试,天擎产品加工系统业务环境和仿真环境容器并发调度性能如图 2 所示:横坐标表示基础软硬件环境,包括 1 种业务环境和 2 种仿真环境;纵坐标表示容器(100 并发)创建所需时间,红色水平线表示业务对容器调度的时效要求上限(2 s)。在业务环境 Intel X86 服务器+Redhat 操作系统时容器(100 并发)创建平均耗时 1.1 s,单个容器创建最大耗时 1.4 s,完全满足业务需要。但在仿真环境耗时较长:华为 ARM 服务器+Kylin 操作系统时平均耗时 8.9 s,单个容器创建最大耗时 20.9 s;海光 X86 服务器+Kylin 操作系统时平均耗时 6.1 s,单个容器创建最大耗时 13.9 s。可见仿真环境的两种架构均不能满足业务时效要求,且存在较大差距。因此需要研究分析仿真环境和业务环境的基础软硬件差异,找到影响容器调度性能的原因,通过优化调整,提高容器并发调度性能,满足业务时效要求。

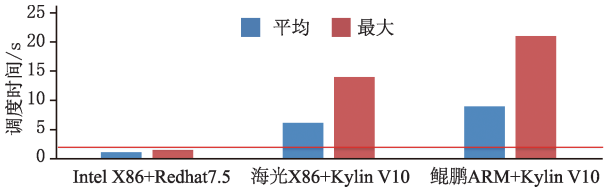

图 2　业务环境和仿真环境容器并发调度性能对比

Fig. 2　Comparison of concurrent scheduling performance for docker between production environment and simulation environment

2 容器调度性能对比测试与分析

为了定位问题原因，本文使用了科学研究中常用的对比分析法，以天擎业务环境和仿真环境的基础软硬件为研究对象，设计了一系列的容器调度性能组合对比测试。由于涵盖 2 个研究对象：基础硬件—服务器和基础软件—操作系统，而每个研究对象又包含若干种选项，因此在设计测试场景时本文固定其中一个研究对象不变，观察另一研究对象在不同选项下的容器调度性能差异。通过以上方法找到所有可满足业务性能要求的基础软硬件组合，并通过对比分析其他组合，找到容器调度性能的影响因素，进而通过调整优化，解决容器调度性能问题。

在基础硬件方面，本文选择了业务环境使用的 Intel X86 服务器，仿真环境使用的华为鲲鹏 ARM 服务器和海光 X86 服务器，具体配置见表 2。在操作系统方面，选择了业务环境使用的 Redhat 操作系统，仿真环境使用的 Kylin 操作系统，因 Redhat 不支持在 ARM 平台安装，增加了 Redhat 的社区版 Centos 操作系统作为参考，共形成 8 个有效测试用例，见表 3。

表 2 测试硬件环境

Table 2 Testing environment for hardware

芯片类型	CPU 类型	CPU 个数	CPU 总核数	内存 GB
Intel X86	Intel_5218_Xeon_2.3GHz	2	32	384
海光 X86	Hygon_C86_7285_2.0GHz	2	64	384
鲲鹏 ARM	Kunpeng920_5220_2.6GHz	2	64	384

表 3 测试场景

Table 3 Test scenario

场景	CPU 架构	操作系统	内核版本	测试用例
1	Intel X86	Kylin V10	4.19.90	是
2	Intel X86	Redhat 7.5	5.10.81	是
3	Intel X86	CentOS 7.6	5.10.81	是
4	海光 X86	Kylin V10	4.19.90	是
5	海光 X86	Redhat 7.5	5.10.81	是
6	海光 X86	CentOS 7.6	5.10.81	是
7	鲲鹏 ARM	Kylin V10	4.19.90	是
8	鲲鹏 ARM	Redhat 7.5	5.10.81	否
9	鲲鹏 ARM	CentOS 7.6	5.10.81	是

2.1 基于服务器的容器调度性能对比

为了对比不同芯片服务器对于容器调度性能的影响，本节分别基于 Kylin、Redhat、Centos 等操作系统开展容器并发调度测试，观察相同操作系统时不同芯片服务器的性能表现。结果如图 3 所示：3 种操作系统均为 Intel 服务器表现最好，海光 X86 服务器次之，鲲鹏服务器最差，且与前 2 种 X86 服务器差距较大。

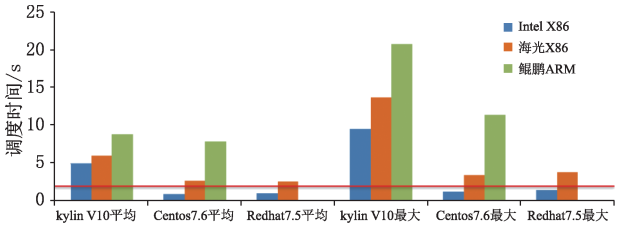

图 3 不同芯片服务器容器并发调度(100 并发)性能对比

Fig. 3 Comparison of concurrent (100) scheduling performance of docker among servers based on different CPU chips

随后对仿真环境的 2 种芯片服务器进行了硬件调优对比测试，方法如下：①鲲鹏 ARM 架构服务器，调整电源模式及调整内存刷新频率。②鲲鹏 ARM 架构服务器，增加内存条数量以开启满通道、调整内存条的安装位置。③海光 X86 架构服务器，开启超线程。

如图 4 所示，海光 X86 架构服务器容器(100 并发)创建平均耗时和单个容器创建最大耗时均与硬件优化前持平，鲲鹏 ARM 架构服务器在搭载 Kylin 操作系统时单个容器创建最大耗时较前缩短，但与业务时效要求仍存在较大差距。可见硬件优化对容器调度性能无明显提升。

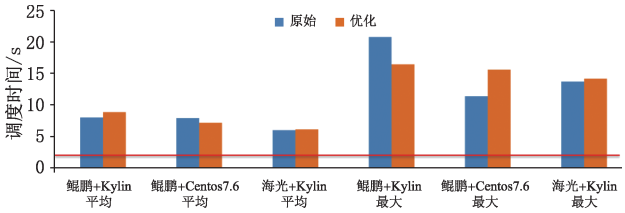

图 4 硬件优化前后容器并发调度(100 并发)性能对比

Fig. 4 Comparison of concurrent (100) scheduling performance of docker before and after hardware optimization

2.2　基于操作系统的容器调度性能对比

根据上述不同芯片服务器及硬件优化的对比测试结果可知,服务器不是影响容器调度性能的主要因素,本节继续分析操作系统对容器并发调度性能的影响。鉴于鲲鹏 ARM 服务器表现与应用性能要求差距较大,重点分析 Intel X86 和海光 X86 服务器搭载不同操作系统时的容器并发调度性能表现。

2.2.1　对比测试

首先分析天擎仿真环境操作系统 Kylin V10 的容器并发调度性能表现,如图 5 所示,在三种芯片服务器下,容器(100 并发)创建平均耗时均在 5 s 以上,最大耗时 10 s 以上,即使 Intel X86 也不能满足 2 s 以内的业务时效要求,且差距较大。

接着分析业务环境 Redhat 系统的容器并发调度性能表现,如图 5 所示,两种 X86 芯片服务器容器(100 并发)创建平均耗时均在 3 s 以内,最大耗时 4 s 以内,Intel X86 完全满足应用时效要求,海光 X86 接近应用时效要求。

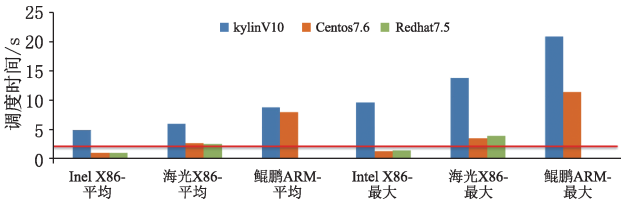

图 5　不同操作系统容器并发调度(100 并发)性能对比

Fig. 5　Comparison of concurrent (100) scheduling performance of docker among operating systems

2.2.2　测试结果分析

如上所述,Kylin 操作系统与 Redhat/Centos 操作系统在容器并发调度性能方面存在较大差距:在相同的硬件环境下 Redhat/Centos 操作系统完全或基本满足业务需求,而 Kylin 操作系统表现与应用需求差距较大,因此下面章节将进一步分析 2 种操作系统的差异。

3　操作系统优化与结果分析

操作系统的差异,主要是操作系统核心—内核的差异,本章进一步对比分析 Kylin 系统和 Redhat 系统的内核差异(Centos 作为 Redhat 的社区版,两者相似性较大,因 ARM 不支持 Redhat 而引入作为参考,业务中并未使用该系统,以下不再赘述)。内核差异主要存在于内核编译选项和内核源代码,本着最小化调整的原则,着重分析内核编译选项。

3.1　内核编译选项差异及分析

操作系统对调度性能的影响与系统响应的实时性和调度策略设置有关,通过对比 Kylin 和 Redhat 内核编译选项在这两方面的变化,发现 2 系统内核编译选项 CONFIG_HZ 和 KERNEL PREEMPTION 设置存在如下差异,见表 4:

表 4　Kylin 系统和 Redhat 系统内核选项设置差异

Table 4　Differences in kernel option settings between Kylin and Redhat

内核选项	Kylin	Redhat
CONFIG_HZ	250	1000
KERNEL PREEMPTION	CONFIG_PREEMPT_NONE	CONFIG_PREEMPT_VOLUNTARY

为了说明上述 2 个内核编译选项对容器并发调度性能的影响,现将其含义及不同设置适用的业务场景解释说明如下:①内核编译选项 CONFIG_HZ。HZ 是操作系统定时器的频率——节拍率(tick rate)的计量单位,在操作系统中,周期性发生的事件都是由系统定时器驱动。内核会根据 HZ 值,设置时钟事件设备,启动时钟中断。HZ 的值越大,系统定时器的精度越高,时钟中断频率越大,系统的响应延迟越小,系统实时性越高。但频繁的时钟中断增大了进程间上下文切换和 cache 更新等次数,增加了系统开销。各操作系统 HZ 默认值根据硬件类型和内核版本有所差异,比如,Kylin V10 操作系统默认配置为 250,而 Redhat 等系统默认值为 1000。该值可通过内核编译选项 CONFIG_HZ 进行设置。针对天擎产品加工系统的特点:同时运行的用户算法数量多,对应需要创建的容器数量多,但算法均是轻量化的,秒级即可完成,因此更大的 HZ 值,有利于提高容器的调度性能,缩小容器创建时间在用户算法总执行时间中的占比。②内核编译选项 KERNEL PREEMPTION。KERNEL PREEMPTION,即内核抢占,Linux 系统从 2.6 内核开始支持内核抢占功能:在内核态允许高优先级的进程抢占低优先级的进程,该功能可有效降低操作系统的响应延时,满足交互式或高实时的应用场景。根据抢占的时机分为可抢占内核模型(PREEMPT)和自愿抢占内核模型(PREEMPT_VOLUNTARY),本文的 Redhat 系统采用的是自愿抢占内核模型,该模

型在内核中选择性地增加了一些抢占点，当低优先级进程在内核态执行时，在预定的抢占点自愿被抢占。目的是减小最大调度延迟和对交互事件提供更快的响应，代价是降低系统吞吐量。相对于可抢占内核模型自愿抢占内核兼顾了系统响应延迟和系统吞吐量。如果希望内核的处理能力最大化，追求最大的系统吞吐量，不考虑调度延迟，可以禁止内核抢占功能，即选择不可抢占内核模型（PREEMPT_NONE），该模型比较适用于科学计算等应用场景。本文的 Kylin V10 操作系统默认设置就是该模型[26-27]。

根据天擎产品加工系统中用户算法的特点：吞吐量不大，基本秒级即可执行完成，而执行算法的容器创建时间如果在 2 s 以上则会明显影响用户算法的执行效率，因此该类型应用相对更大的系统吞吐量，更追求较低的系统响应延迟，但又不需要工业级嵌入式系统毫秒级的实时性，因此选择自愿抢占内核模型可以很好的做到系统响应延时和系统吞吐量的平衡。

3.2 内核编译选项调整及结果

根据上述分析，Redhat 系统的内核编译选项设置更适合天擎仿真环境的应用场景，因此在 Kylin 系统原生内核基础上，移植采用 Redhat 系统的两个内核选项参数，即设置：①内核抢占策略为“自愿内核抢占”：CONFIG_PREEMPT_VOLUNTARY＝y。②内核节拍率为“1000Hz”：CONFIG_HZ_1000＝y CONFIG_HZ＝1000。

调整后对 Kylin 系统内核进行重新编译，生成的系统本文称为 Kylin 系统（新内核），与 Kylin 系统（原生内核）以及 Redhat 系统进行对比测试。如图 6 所示，调整内核选项后的 Kylin 系统（新内核）容器并发调度性能较 Kylin 系统（原生内核）大幅度提升：容器（100 并发）创建平均耗时由 6 s 降至 2 s 以内，单个容器创建最大耗时由 13 s 降至 2 s 以内；同时在海光 x86 平台，对比 redhat 系统性能也有了进一步提升：容器（100 并发）创建平均耗时由 2.6 s 降至 1.2 s，单个容器创建最大耗时由 3.9 s 降至 1.9 s，调整内核编译选项后的 Kylin 操作系统无论在 Intel X86 平台还是海光 X86 平台均可完全满足应用时效要求。

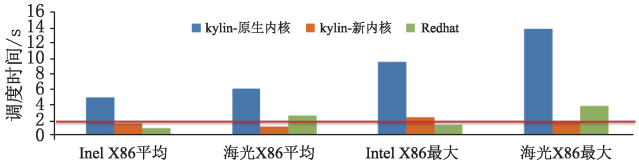

图 6 Kylin 系统内核选项调整后容器并发调度（100 并发）性能对比

Fig. 6 Comparison of concurrent (100) scheduling performance of docker before and after adjusting KylinOS kernel options

4 结论

本文针对天擎仿真环境容器并发调度性能无法满足产品加工系统时效要求的问题，开展了一系列的组合对比测试，分析了操作系统内核设置对系统实时性和吞吐量的影响以及适用的业务场景，通过调整 Kylin 操作系统的内核选项设置，使得仿真环境容器调度性能得到大幅提升，解决了天擎业务在仿真环境运行的性能问题。

本文的研究工作为国家级和省级天擎系统向国产平台迁移提供了指导：在服务器硬件方面，建议选用性能更高的海光 X86 芯片服务器。在操作系统设置方面，建议适度提高系统的实时性，即适度增大内核定时器的节拍率，采用内核自愿强占策略。同时也为其他行业在国产平台使用容器技术提供了经验。

参考文献

[1] 王甫棣，王帅，赵希鹏. 基于 ElasticSearch 的异构数据迁移方法[J]. 气象科技，2022，50(1)：45-46. Wang F D, Wang S, Zhao X P. A method for heterogeneous data migration based on ElasticSearch [J]. Meteorological Science and Technology (in Chinese), 2022, 50(1): 45-46.

[2] 刘媛媛，何文春，王妍，等. 气象大数据云平台归档系统设计及实现[J]. 气象科技，2021，49(5)：697-706. Liu Y Y, He W C, Wang Y, et al. Design and implementation of meteorological big data cloud platform archive storage system [J]. Meteorological Science and Technology (in Chinese), 2021, 49(5): 697-706.

[3] 韩同欣，何文春，吴鹏，等. 气象基础设施云平台关键设计与实现[J]. 气象科技，2022，50(2)：290-296. Han T X, He W C, Wu P, et al. Key design and implementation of cloud platform for meteorological infrastructure [J]. Meteorological Science and Technology (in Chinese), 2022, 50(2): 290-296.

[4] 王怀乐，刘然，马明，等. 基于云资源的青藏高原科考数据收

集与传输平台设计与实现[J]. 气象科技，2023，51(5)：648-657.　Wang H L，Liu R，Ma M，et al. Design and implementation of data collection and transmission platform for Qinghai Tibet Plateau scientific research based on cloud resources [J]. Meteorological Science and Technology (in Chinese)，2023，51(5)：648-657.

[5] 徐迅，李万乐，骆汉宾，等. 建筑企业 BIM 私有云平台中心建设与实施[J]. 土木工程与管理学报，2014，31(2)：7.　Xu X，Li W L，Luo H B. Construction and implementation of BIM private cloud platform in construction enterprises [J]. Journal of Civil Engineering and Management (in Chinese)，2014，31(2)：7.

[6] 肖建华，王厚之，彭清山，等. 地理时空大数据管理与应用云平台建设[J]. 测绘通报，2016，4：38-42.　Xiao J H，Wang H Z，Peng Q S，et al. Research on the construction of cloud platform for the spatial-temporal big data management and application [J]. Bulletin of Surveying and Mapping (in Chinese)，2016，4：38-42.

[7] 程平，万家盛. 大数据时代财务共享服务中心云平台的构建及其应用[J]. 商业会计，2015，15：20-22.　Cheng P，Wan J S. Construction and application of cloud platform for financial shared service center in the Big Data Era [J]. Commercial Accounting (in Chinese)，2015，15：20-22.

[8] 江彩英，郭晓佳，谢丹，等. 基于虚拟化云平台的气象终端集约化管理[J]. 气象科技，2014，42(5)：5.　Jiang C Y，Guo X J，Xie D，et al. Intensive management of meteorological terminals based on virtualized cloud platform [J]. Meteorological Science and Technology (in Chinese)，2014，42(5)：5.

[9] 王斌，李楠，孙月新，等. 国家教育云平台架构设计及关键技术分析[J]. 现代电信科技，2013，43(5)：4.　Wang B，Li N，Sun Y X，et al. Architecture design of the state education cloud platform and analysis of key technologies [J]. Modern Science & Technology of Telecommunications (in Chinese)，2013，43(5)：4.

[10] 张宝利，朱月. 全媒体融合云平台设计[J]. 广播与电视技术，2015，42(2)：3.　Zhang B L，Zhu Y. Design of omni-media cloud platform [J]. Radio & TV Broadcast Engineering (in Chinese)，2015，42(2)：3.

[11] 刘克浩，肖飞龙. 基于云平台和大数据的新型健康管理模式[J]. 公共卫生与预防医学，2014，25(5)：89-91.　Liu K H，Xiao F L. Novel health management model based on cloud platform and big data [J]. Journal of Public Health and Preventive Medicine (in Chinese)，2014，25(5)：89-91.

[12] 罗亮，吴文峻，张飞. 面向云计算数据中心的能耗建模方法[J]. 软件学报，2014，25(7)：1371-1387.　Luo L，Wu W J，Zhang F. Energy modeling based on cloud data center [J]. Journal of Software (in Chinese)，2014，25(7)：1371-1387.

[13] 彭勇，谢剑，童遥，等. 一种基于 Docker 的数据中心云平台实现方法及系统[J]. 中兴通讯技术，2017，23(2)：3.　Peng Y，Xie J，Tong Y，et al. A cloud platform for data center based on Docker [J]. ZTE Technology Journal (in Chinese)，2017，23(2)：3.

[14] 王亚玲，李春阳，崔蔚，等. 基于 Docker 的 PaaS 平台建设[J]. 计算机系统应用，2016，25(3)：72-73.　Wang Y L，Li C Y，Cui W，et al. Construction of Docker-based PaaS [J]. Computer Systems & Applications (in Chinese)，2016，25(3)：72-73.

[15] 杨鹏，马志程，彭博，等. 基于 Docker 容器的 OpenStack 云平台性能研究[J]. 计算机工程，2017，43(8)：26-27.　Yang P，Ma Z C，Peng B，et al. Performace research of OpenStack cloud platform integrated with Docker container [J]. Computer Engineering (in Chinese)，2017，43(8)：26-27.

[16] 林伟伟，齐德昱. 云计算资源调度研究综述[J]. 计算机科学，2012，39(10)：6.　Lin W W，Qi D Y. Survey of resource scheduling in cloud computing [J]. Computer Science (in Chinese)，2012，39(10)：6.

[17] 秦志光，柯涛，刘梦娟，等. 面向云平台的资源分配策略研究[J]. 计算机应用，2013，33(2)：299-303.　Qin Z G，Ke T，Liu M J，et al. Resource allocation strategies for cloud platform [J]. Journal of Computer Applications (in Chinese)，2013，33(2)：299-303.

[18] 王吉，包卫东，朱晓敏. 虚拟化云平台中实时任务容错调度算法研究[J]. 通信学报，2014，35(10)：11.　Wang J，Bao W D，Zhu X M. Fault-tolerant scheduling algorithm for real-time tasks in virtualized cloud [J]. Journal of Communications (in Chinese)，2014，35(10)：11.

[19] 王小雪，王晓锋，刘渊. 基于 OpenStack 的高资源利用率 Docker 调度模型[J]. 计算机工程，2022，48(9)：171-179.　Wang X X，Wang X F，Liu Y. OpenStack-based Docker scheduling model with high resource utilization [J]. Computer Engineering (in Chinese)，2022，48(9)：171-179.

[20] 刘哲源，吕晓丹，蒋朝惠. 基于模拟退火算法的粒子群优化算法在容器调度中的应用[J]. 计算机测量与控制，2021，29(12)：177-183.　Liu Z Y，Lv X D，Jiang C H. Application of particle swarm optimization algorithm based on simulated annealing algorithm in container scheduling [J]. Computer Measurement & Control (in Chinese)，2021，29(12)：177-183.

[21] 邓广宏，张棋恒. 基于混合关键系统的容器调度架构设计[J]. 计算机科学，2023，50(S01)：901-915.　Deng G H，Zhang Q H. Container-based scheduling architecture for mixed-criticality systems [J]. Computer Science (in Chinese)，2023，50(S01)：901-915.

[22] 姚卫红，黄小远，方仁孝. 基于车联网应用的云平台任务调度算法[J]. 计算机仿真，2014，31(10)：165-169.　Yao W H，Huang X Y，Fang R X. Cloud platform task scheduling algorithm based on Internet of Vehicles [J]. Computer Simulation (in Chinese)，2014，31(10)：165-169.

[23] 张玉清，王晓菲，刘雪峰，等. 云计算与环境安全综述[J]. 软件学报，2016，27(6)：1328-1348.　Zhang Y Q，Wang X F，

Liu X F, et al. Survey on cloud computing security [J]. Journal of Software (in Chinese), 2016, 27(6): 1328-1348.

[24] 冯登国，张敏，张妍，等. 云计算安全研究[J]. 电子技术与软件工程，2011，22(1)：71-83. Feng D G, Zhang M, Zhang Y, et al. Study on cloud computing security [J]. Electronic Technology & Software Engineering (in Chinese), 2011, 22 (1): 71-83.

[25] 杨保华，戴王剑，曹亚仑. Docker 技术入门与实战[M]. 北京：机械工业出版社，2018. 162. Yang B H, Dai W J, Cao Y L. Introduction and practice of Docker technology (in Chinese) [M]. Beijing: China Machine Press, 2018. 162.

[26] Evi N, Garth S, Trent R H, et al. UNIX And Linux System Administration Handbook [M]. 4th ed. Britain: Prentice Hall, 2010: 568-591.

[27] Daniel P B, Marco C. Understanding the Linux Kernel [M]. 3rd ed. America: O'Reilly Media, 2005: 137-159.

Research on Optimization of Docker Scheduling Performance for Simulation Environment of Meteorological Big Data Cloud Platform

WU Peng[1] HAN Tongxin[1] CHEN Shiwang[1] NIE Yuanding[1] ZHENG Xiaozhi[2]

(1 National Meteorological Information Center, Beijing 100081; 2 Haizhu Bureau of Meteorology, Guangzhou 510240)

Abstract: In order to achieve the goal of independent and controllable key core technologies for Meteo by 2025, the Meteo Big Data Cloud Platform (referred to as Tianqing) establishes a simulation environment based on Hygon X86 CPU and Kylin OS. However, in the operation of simulation platforms, it finds that the docker scheduling performance of data processing and assembly line subsystems based on Kubernetes is poor, which cannot meet the timeliness requirements of user integration algorithms. In response to this issue, this article adopts a comparative analysis method, selecting servers based on three types of CPU and three types of operating systems from the simulation environment and business environment for Tianqing as the research objects. A series of combined comparative test cases are designed. It finds that the kernel is the key factor affecting docker scheduling performance. Further analysis is conducted on the impact of operating system kernel settings on real-time and throughput, as well as the suitable business scenarios. Finally, a method for adjusting the Kylin OS kernel is provided. By adjusting the kernel settings, the docker scheduling performance significantly improves, meeting the timeliness requirements of the data processing system and laying the foundation for achieving self-supporting of the key core technology of Tianqing.

Keywords: Meteo Big Data Cloud Platform; self-supporting; Kylin OS; docker scheduling

Received: 2023-6-2 Accepted: 2024-1-12

© Editorial Office of *Meteorological Science and Technology* (CC BY-NC-ND 4.0)