



# 华中科技大学

## 云计算与虚拟化课程报告

姓名： 练炳诚

班级： CS1705 班

学号： U201714707

2020 年 6 月 19 日

## 所选文献

- 名称: A Performance Study of Containers in Cloud Environment.

- 作者: Bowen Ruan, Hang Huang, Song Wu, Hai Jin.

to appear in Proceedings of the 10th International Conference on Asia-Pacific Services Computing (APSCC 2016), Zhangjiajie, China, November 16-18, 2016, pp. 343-356.

- 链接: [http://cgcl.grid.hust.edu.cn/wusong/file/APSCC16\(Ruan\).pdf](http://cgcl.grid.hust.edu.cn/wusong/file/APSCC16(Ruan).pdf)

## 报告内容

本报告分析了文献的问题提出、解决方案、提出解决方案的动机、个人见解，最后是对课程的总结。

# 1.论文的提出

## 1.1 容器背景

由于容器在云环境中可以提供接近于本地的性能，因此容器技术非常受欢迎。根据不同的设计目的和底层实现，容器可以分为应用程序容器(如 Docker)和操作系统容器(如 LXC)。容器的多样性可能导致在选择哪种容器以适合不同的使用场景时出现争议。与此同时，由于云平台倾向于在虚拟机中运行容器，因此公共容器服务的架构备受争议。从性能的角度来看，裸机和容器之间额外的虚拟机层可能会带来不必要的性能开销。

容器技术相较于虚拟机技术，具有更快的启动速度和更小的内存占用，它们之间的关键差别在于是否拥有独立的系统内核，图 1 展示了两者的架构差异。

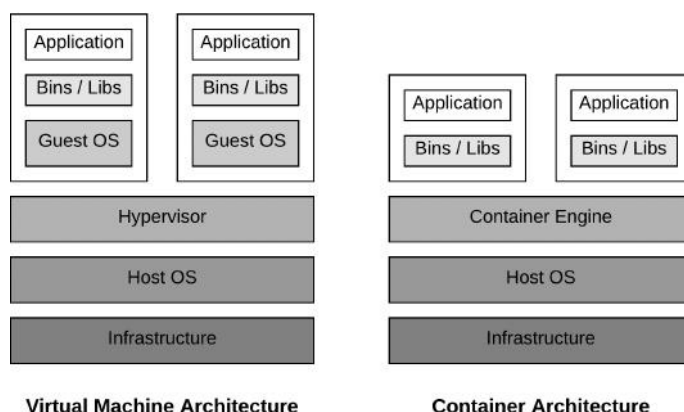


图 1 虚拟机和容器的架构差异

作为一个系统架构，基础服务设施和主机操作系统是两者都具备的，不同之处在于虚拟机基于虚拟机监管程序，建立相互隔离的虚拟操作系统作为客户机操作系统，应用程序在这些操作系统上运行；而容器是基于容器引擎运行应用程序，无需建立客户机操作系统，所有应用程序共享主机内核。

容器按技术可分为应用程序容器和操作系统容器，应用程序容器只包含单个进程，并在进程结束后停止该容器，其思想是将应用程序分解到多个容器中运行，提供了分层文件系统。而操作系统容器包含完整的运行时环境，其思想是在容器中提供完整的操作系统环境，更像是轻量级的虚拟机，提供了中立的文件系统，主要提供基础服务设施。

## 1.2 研究目的

由于部署应用程序的便携性，容器技术在云平台上引发了一次压倒性的革命。与过去相比，开发人员不再需要花费很长时间来安装软件和调整配置，他们可以简单地从 Docker hub 中提取 image 并启动容器。因此，公共容器服务是开发人员以高效的方法构建、部署和运行应用程序的新解决方案。

但是现有公共容器服务的一个主要缺陷是云平台倾向于在虚拟机中运行容器，以获得虚拟机中成熟的管理工具的支持。显然，在裸机和容器服务之间添加额外的虚拟机层可能会产生不必要的性能开销。基于当前行业如何准确地度量虚拟机层的开销，探讨如何合理地提供容器服务，是一个值得全面研究的问题。

## 2. 解决方案

文献中进行了两个部分的评价，第一个部分是对容器技术的评估，第二个部分是对公共容器服务平台的评估。在容器技术评估部分，对 LXC、Docker，以及通过在虚拟机上安装 Docker 测试 Docker-Machine 的性能这三个不同形式的容器进行研究。在公共容器服务平台的评估部分，测试了 Redis、MySQL 在 ECS(Amazon EC2 Container Service)、GKE(Google Container Engine)上的性能，以及在 ECS、GKE 上分别测试了 Hadoop 和 Spark 的性能。

文献中选取的测试环境是一台拥有 32 核 Intel X5650 CPU 和 64GB 内存的服务器，搭载 Ubuntu 15.10 操作系统，运行 Linux 4.2 内核，LXC 版本 1.1.3，Docker 版本 1.9.1。对不同的容器，通过设置 CGroups 参数将容器的资源消耗限制在同一水平，每个容器拥有两个 vCPU 和 8GB 内存，对 Docker-Machine，创建了具有两个 vCPU 和 8GB 内存的虚拟机。此外，还搭建了一个包含 8 个计算节点的容器集群来测试分布式应用程序的性能，每个节点同样分配两个 vCPU 和 8GB 内存。硬件方面，选择了 m4.large 实例作为标准计算节点，对于 GKE(Google Kubernetes Engine)，用 EC2 定制等效规范的实例。以上配置尽可能的保证了公平性。

## 2.1 容器技术评估方案及结果

在容器技术的评估中，文献主要从 CPU 性能、内存带宽性能、磁盘 I/O 性能以及网络延迟 4 个方面进行了评估。

在 CPU 性能上，选取了 450.soplex 和 473.astar 两个业界标准化的基准测试套件，进行了整数和浮点运行能力的评估，测试结果表明 LXC 和 Docker 在 CPU 性能上相当，均比 Docker-Machine 高，说明额外的虚拟机层会阻碍 Docker 的性能。

在内存带宽性能上，选取 Stream 作为内存基准测试程序，结果表明 LXC、Docker、Docker-Machine 在内存性能上没有显著的差异，说明即使添加了虚拟机层，也只会造成微不足道的开销。

在磁盘 I/O 性能上，采用了 fio 作为基准测试程序，结果表明 LXC 在随机读和随机写上性能最好，分别高出 Docker 6.1%和 16.6%，而 Docker-Machine 性能损失最严重，分别低于 Docker 40.0%和 42.7%，文献指出 Docker 在磁盘 I/O 性能上的损失是由于其默认文件系统 AUFS 造成的，AUFS 架构中文件系统分为 Image 层和 Container 层，Image 层是只读层，仅保存相对底层分支的差异，Container 层是只写层，保存修改，AUFS 在写性能会造成较大的延迟，因为容器第一次写任何文件时，必须找到文件并将其复制到容器的顶部可写层，AUFS 分支中的文件搜索和将文件复制到顶层可写层的要求导致 Docker 额外的磁盘 I/O 延迟。

在网络延迟性能上，使用了 Netperf's request-response 模式进行测试，测试结果显示，为 Docker 的网络延迟是 LXC 的 1.25 倍，而 Docker-Machine 的延迟高达 3 倍多，由于测试是在局域网内进行，传输延迟可忽略，Docker-Machine 的高延迟主要是由处理延迟决定的，文献指出 Docker 相对于 LXC 增加了 NAT 映射机制，而在 Docker-Machine 模式下，网络包(Network Packets)通过虚拟设备由客户机 OS 传向主机 OS，其硬件仿真和 NAT 映射机制造成了较高的延迟。

## 2.2 公共容器服务评估方案及结果

该部分对 Redis 和 MySQL 在 ECS、GKS 上的性能以及分布式数据处理系统 Hadoop、Spark 在 ECS、GKE 上的性能进行了评估。

对 Redis，由于其是内存数据库，故网络延迟是其主要关注点，文献采用了 Redis-benchmark 作为基准测试程序，结果显示其性能呈现为 LXC > Docker > ECS > GKE > Docker-machine，ECS 的网络延迟相对 GKS 更小，性能更好。

对 MySQL，这是一个开源的，将数据在磁盘上持久保存的数据库，因此磁盘 I/O 性能决定了 MySQL 的性能。文献采用了 SysBench 作为基准测试程序，在数据库中产生 1000000 条记录，然后将工作线程从 1 增加到 16，按只读模式和混和读写模式查询数据库，记录了每种情况下每秒执行事务的数量，结果显示，在只读模式下，LXC 和 Docker 相对 ECS、GKE 均有更好的性能，而在混合读写模式下，ECS、GKE 拥有更好的性能，文献对这种产生这种现象的原因推测为 ECS 和 GKE 厂商的机器使用了 SSD 作为存储介质，而本地搭载容器的机器使用的是 HDD。此外，线程数量较少时，ECS 的性能高于 GKE，随着线程数量的增加，GKE 的性能逐渐高于 ECS，文献对这一现象的原因推测为 ECS 厂商提供的实例计算能力较强，而 GKE 厂商提供的实例使用的 SSD 的 I/O 性能优于 ECS。

在分布式数据处理系统的性能评估上，文献采用了四种基准测试程序，包括 WordCount, TeraSort, PageRank 和 Kmeans，均是计算密集型工作负载程序，测试结果表明，Hadoop 和 Spark 系统在 ECS 平台的性能均在不同程度上优于 GKE，且远远优于 Docker-Machine，但与 LXC 和 Docker 均有差距，这也说明了在裸机和容器间构建虚拟机层会造成额外开销，此外，结果表明 LXC 的性能高于 Docker，文献指出由此前的实验结论，LXC 和 Docker 的 CPU 性能相当，但磁盘 I/O 性能却是 LXC 优于 Docker，当有大量 I/O 请求时，Docker 会出现较大的 I/O 延迟。

### 3.提出解决方案的动机

从文献中了解到，容器技术正越来越受到学术界的重视，许多的研究工作都对容器和虚拟机进行了广泛的比较，但这些研究没有讨论不同容器之间的区别，以及在裸机和容器之间添加额外虚拟化层的影响。

对云平台的评估可以为开发人员选择合适的平台部署应用程序和节约成本提供有价值的指导，在现有的一些研究基础上，通过对公共云服务平台如 ECS、GKE 的服务质量进行评估，能够提供一些有价值的指导。

### 4.对文献的个人见解

本篇文献主要对应用程序容器(Docker)和操作系统容器(LXC)的差异进行了讨论，以及对裸机和容器间添加虚拟机层的影响进行了探究，最后对公共容器服务平台中的 ECS 和 GKE 在 Redis 和 MySQL、Hadoop 和 Spark 下的服务质量进行了比较。文献中对容器技术的性能评估和公共容器服务平台的性能评估使用了大量基准测试程序，为保证实验的公平性，对测试环境进行了严格的变量控制，尽可能使得不同容器运行时分配的系统资源相同，同时对同一项测试进行了多次实验消除偏差，实验过程较为严谨。

通过本篇文献的阅读，我了解到了应用程序容器、操作系统容器和虚拟机三者的差别，虚拟机体格较为庞大，提供的功能和工具支持最多，但也会造成较大的开销，操作系统容器更像是一个轻量级的虚拟机，而应用程序容器就更加轻巧。在实验方案中对性能的评估上，主要分析了 CPU 性能、内存性能、磁盘 I/O 性能以及网络带宽性能，这些是影响容器性能的主要因素，得到的结论是操作系统容器在处理高磁盘负载的应用程序上更有优势，适合于提供底层基础服务设施，应用程序容器更适合于运行微服务和分发应用程序。通过 Docker-Machine 的表现能够看出在裸机和容器间增加虚拟机层会带来严重的磁盘 I/O 开销和网络延迟，这主要是由于中间虚拟机层的传输开销和硬件仿真造成的。

在对公共容器服务平台的对比中，有两个实验结果是在我意料之外的，其中一个就是在测试 MySQL 数据库时，只读模式下本地容器的性能都优于 ECS、GKE，而在读写混合模式下反而 ECS、GKE 的性能更好，文献分析原因是由于

公共容器平台的实例使用了 SSD 作为存储介质，而本地机器使用的是 HDD。通过上网查阅资料了解，一般 HDD 的顺序读性能高于其写性能，在只读模式下，本地 HDD 无需通过中间虚拟机层就可直接访问数据，而公共容器服务平台虽然使用了 SSD，但由于存在网络延迟和中间虚拟机层带来的开销，导致实际测试出来的结果是本地 HDD 的读性能更好，但换成混合读写模式时，由于存在写操作，而 HDD 的写性能相对其读性能更低，此时 SSD 的读写速度优势就体现出来了，就导致公共容器服务平台的性能高于本地容器，我认为如果实验时能尽可能将本地硬件设施配置为和公共容器服务平台提供的实例的硬件设施相同，对在裸机和容器间添加虚拟机层会造成额外开销的这一结论更有说服力。

另一个实验结果是在只读模式和混合读写模式下，ECS 和 GKE 的性能差异随线程的增加而改变，线程少时 ECS 性能更好，线程多时 GKE 性能更好，文献推测原因是 ECS 实例的计算能力更强，而 GKE 实例的磁盘 I/O 更快，并为进行更深层次的探究。

个人认为，在对 MySQL 数据库中同一批数据进行读写时，磁盘 I/O 的总量和计算任务的工作量是一定的，线程的多少决定了磁盘 I/O 的次数和计算任务的次数，由于操作系统内核采用了 I/O 调度方案，线程对磁盘的 I/O 请求会写入 I/O 队列，线程数量增多时，如果这些线程访问的数据在磁盘中是相邻的，那么 I/O 调度队列里面的 I/O 请求数据也是相邻的，随着线程数量增加，这样的顺序性会更明显，如果是磁盘介质，那么由于访问的是连续存储区，就会显著减少磁盘寻道时间，但如果是如上一个结论中说明的这些公共容器服务平台使用了 SSD 存储介质，就不存在磁盘寻道时间的问题，因此多个线程带来的这种减少磁盘寻道时间的优势也就不存在。另一方面，在计算量一定时，线程数量增多会带来更多的调度时间开销，相对而言较高的计算性能在线程数量增大时优势会减弱，这符合文献对 ECS 实例计算性能高于 GKE 实例的推测。由于文献的出发点是通过进行性能评估为科研和开发人员在部署应用程序的选择上提供有价值的指导，因此我觉得此处未进行深入的探究稍显遗憾，且公共容器服务平台每年会不断更新其提供的服务机的硬件配置，仅是对当时的性能进行评估不能起到指导作用，如果能提供一套评估方案，即此处进行深入探讨，将更具指导价值。



以上就是我在阅读完文献后的一些总结和思考,由于我在之前从未对相关领域的研究工作进行了解,再加上对本篇文献的研究工作了解度不够,以及自身知识能力的不足,可能见解中存在错误,恳请老师指正。

## 5.课程总结

通过课程学习和自己从网上的学习,了解了云计算的产生背景,随着大数据时代的到来,业界对数据处理的要求也越来越高,但是个人或企业购买高性能的计算机和服务软件会造成较大的成本开销,这些原因促使着资源共享和计算能力的分布式共享的产生,从而促使着云计算的产生。知道了目前主流的一些云计算服务提供商,如亚马逊、谷歌、阿里云等,以及云计算的部署模式:私有云、公有云、混合云的特点和区别,了解了云计算数据中心背后的能耗问题,其中供电系统和散热系统起到重要作用,散热系统中如 Apple 利用了贵州四季气候凉爽的特点将数据中心建立在贵州, Goole 将数据中心建立在海洋中利用水进行散热,供电系统中 Google 利用风能这类绿色能源进行供电。

接着学习了云计算中的虚拟化技术,虚拟化技术就是利用软件对物理固件如 CPU、内存等进行仿真,从而将一个主机的物理资源映射为多个虚拟资源供使用,包括服务器虚拟化、存储虚拟化、网络虚拟化和桌面虚拟化等,虚拟机迁移技术是云计算的关注重点,关乎到云计算的实用性,而考虑到数据安全,虚拟机的隔离技术能够使得不同虚拟机之间保存独立性。

之后学习了云计算服务的四层模型:应用层、平台层、基础设施层、虚拟化层,分别对应的服务:软件即服务(SaaS)、平台即服务(PaaS)、基础设施即服务(IaaS)以及硬件即服务,其中虚拟化层结合 PaaS 一起提供硬件服务,即主要分为 SaaS、PaaS、IaaS。

然后重点学习了 Docker 容器技术, Docker 是构建在 LXC 上,基于进程容器的轻量级虚拟化方案,通过容器引擎,把 Linux 的 namespace、cgroup 技术进行封装,供统一使用,运行在容器中的应用程序共享内核,相较于虚拟机架构开销很小,启动速度快,能够提供近乎本机的性能,且容器本身具有隔离性,能让开发者先在本机上构建一个复杂的程序或架构来进行测试,而不是一开始就在生产环境部署、测试。

最后，还了解了一些开源分布式数据处理系统，如 Hadoop 和 Spark 以及这两者的应用。

受今年疫情的影响，我在家里度过了大学最繁忙的一个学期，本以为会对课程学习有很大的影响，但是学院通过组织线上授课的方式还是能按时学习课程，虽然少了线下和同学们的交流，但线上上课也可以通过录屏课后进行复习巩固。通过这门课程的学习，也让我了解到了当今云计算的发展情况和一些基本技术，但课程内学到的内容仅仅只是皮毛，希望能以此作为今后更深一步学习的敲门砖。