基于 Docker 的 Hadoop 平台架构研究

赵乐乐，黄 刚，马 越
（南京邮电大学 计算机学院，江苏 南京 210003）

摘 要：Hadoop 作为云计算中重要的大数据处理平台，需要较高的读写速率。传统的虚拟化技术对于物理主机的资源利用率，无法达到真实物理主机的水平。同时，传统虚拟化技术难以灵活配置文件和自动化创建、部署机制。容器是基于共享 Linux 内核的一种虚拟化技术，能够达到接近物理主机的资源利用率。Docker 是一种轻量级新兴的虚拟化容器技术，在复杂的集群系统的搭建方面，具有可移植、易使用、跨平台等优势。所以，在复杂的分布式应用集群的部署中，Docker 能够快速、准确、标准化封装应用程序并自动化部署整个运行环境。因此，Docker 是容器虚拟化技术下一个相对成熟的实现方案。通过实验验证了 Docker 相比传统虚拟化技术在读写性能上的优势，并构建了基于 Docker 的 Hadoop 平台，讨论了在 Docker 上构建 Hadoop 的优势。

关键词：Hadoop；虚拟化；容器；Docker
中图分类号：TP31 **文献标识码**：A **文章编号**：1673－629X(2016)09－0099－05
doi：10.3969/j.issn.1673－629X.2016.09.023

Research on Hadoop Platform Based on Docker

ZHAO Le－le，HUANG Gang，MA Yue
(School of Computer，Nanjing University of Posts and Telecommunications，Nanjing 210003，China)

Abstract：Hadoop，as an important big data processing platform，needs higher I/O rate. For the resource utilization of physical host，the traditional virtualization technology cannot reach the level of real physical host. Meanwhile，it is difficult to configure the files flexibly and create and deploy mechanisms automatically. The container is a virtualization technology based on sharing Linux kernel，which can reach the resource utilization close to the physical host. Docker emerging is a lightweight container of virtualization technology，and in the complex cluster system construction，it is portable and easy to use，with cross－platform. So，in the complicated distributed deployment of application clusters，Docker can be rapid，accurate，and standardized packaged applications and deploy automatically whole runtime environment. Therefore，Docker is one of the mature implementation scheme of the container virtualization technology. It is verified by the experiment that the Docker is better than traditional virtualization technology in reading/writing performance，and the Hadoop platform based on the Docker is established and the advantage of Hadoop on Docker is discussed.

Key words：Hadoop；virtualization；container；Docker

0 引 言

Hadoop 作为一款优秀的 PaaS 软件，能够让开发者更有效率的开发与部署，而不需关注底层。对于 Hadoop 的部署，一般来说有两种选择：一种是直接部署在物理主机上，一种是部署在虚拟机上。当然，大多数时是根据情况，混合使用物理主机和虚拟机来部署 Hadoop。对于商业公司来说，一般都会把 Hadoop 部署在 IaaS 提供的虚拟机上，以提高对单点故障等问题的处理速度，利于运维人员的管理维护，而不是直接部署在物理主机上。然而，由于虚拟机本身需要消耗一定资源来实现底层的虚拟化，所以势必会消耗物理主机的资源。

对于企业级的 Hadoop 部署，底层的虚拟化方案组成有很多，当前业界提供的虚拟机方案大致可分为三类：XEN、KVM 以及 Vmware。对于企业应用来说，可能会选择使用混合多种虚拟机方案的方式，也有很多厂商专门提供组合虚拟化技术的解决方案。但无论这些虚拟化解决方案是如何组成的，它们都可归类为使用全虚拟化技术或使用部分虚拟化技术，这些虚拟化技术都要提供一个完整的虚拟硬件环境，从而能

收稿日期：2015－12－03 **修回日期**：2016－04－06 **网络出版时间**：2016－08－01
基金项目：国家自然科学基金资助项目(61171053)；南京邮电大学基金(SG1107)
作者简介：赵乐乐(1992－)，男，研究方向为计算机云计算与大数据应用；黄 刚，教授，研究方向为计算机软件理论及应用。
网络出版地址：http://www.cnki.net/kcms/detail/61.1450.TP.20160801.0909.072.html

让一个操作系统完整地安装在虚拟机中。现在各大云计算虚拟化厂商的 IaaS 层，无论是 AWS、Google 还是国内的 BAT 公司，基本都使用这种虚拟主机的方式。这种方式有其自身的优势，即能够充分利用物理机的资源，能够自由选择所要创建的主机的虚拟硬件，能够构建各种系统环境。但由于每台虚拟机都是完整地安装整个操作系统，虚拟机之间不能共享相同的底层操作系统的功能，并且在实际使用过程中，往往并不需要完整的操作系统支持，只需要其中的一些软件，只是因为安全与维护等因素才需要虚拟机。所以，一个完整的操作系统虽然功能强大，但也造成了一定的资源浪费。正是基于这一原因，近年来出现了新的轻量级的虚拟化技术—容器。

1 容器技术

文中所指的容器技术是 Linux 容器(Linux Container，LXC)技术。容器有效地将由单个操作系统管理的资源划分到孤立的组中，以便更好地在孤立的组之间平衡有冲突的资源使用需求[1]。容器的思想就是把一个进程(包括其子进程)独立打包在一个“集装箱”内，而不影响系统内的其他进程。这样做的最大好处就是，容器内的进程所运行的环境是独立于底层物理操作系统的，当容器启动的时候，也仅是通过进程间调度，而不需要引导整个系统，从而提高效率。

容器技术在 Linux 的 2.6.29 版本中就产生了，并且已有多种实现方案，但之前的技术并没有引起业界的关注。2013 年 Docker Inc 公司发布了一个开源的基于 LXC 技术的容器引擎，使得容器技术获得了业界的重视。现在 Docker 技术已经被 Google、IBM、RedHat、Amazon 和微软等业界知名公司所支持。虚拟机和容器正在改变运行、设计、开发和部署应用程序的方式。

Docker 受欢迎的原因是其主要解决了以下问题：

(1)快速部署。通常一款产品能够成功发布，开发者需要关心很多东西，从操作系统到中间件再到应用，而且这些东西都难于管理，这个问题在软件行业普遍存在。Docker 则简化了这些工作，比如 Web 应用、后台应用、数据库应用、大数据应用，Hadoop 集群、消息队列等都可以打包成一个镜像，然后再快速部署。

(2)软件管理。亚马逊的 AWS 之所以成功，是因为它让开发者将应用转移到云上，解决了硬件管理的问题，然而软件配置和管理相关的问题依然存在。Docker 正好能帮助软件开发者使用新的软件管理方法来解决这个问题。

(3)虚拟化技术的改变。云计算使用标配的硬件来降低成本，采用虚拟化的手段来满足用户按需分配的资源需求以及保证可用性和隔离性。然而无论是 KVM 还是 Xen，都存在一定的资源浪费，因为用户需要的是高效的运行环境而非一个操作系统，虚拟主机既浪费资源又难于管理，轻量级的 LXC 则更具灵活性和快速性。

(4)更高的便携性。LXC 在 Linux 2.6 的内核里就已经存在了，但是其设计之初并不是为云计算考虑的，缺少标准化的描述手段和容器的可便携性，决定其构建出的环境难于分发和标准化管理。Docker 则在这个问题上做出了实质性的创新方法[2]。

从模块化和整合化的方面来说，可以认为 Docker 被设计的初衷是在独立封装和在任何平台都可以同步运行。Docker 将操作系统、虚拟机、物理机和基于上面的操作整合起来进行商品化，同时提供了一系列的 API，使得其他人能够基于这些 API 进行操作。但 Docker 不能商品化的部分是数据中心[3]，这是其目前的缺陷。

从一个开发者的角度，把应用封装在 Docker 的意义在于，可以把整个云服务作为一个模块进行操作，这其中的模块只是一个可以被替代的商品。Docker 的优势在于，可以任意地把应用进行迁移而无需做出其他改动。所以对于开发者而言，封装应用只需要 Docker 就够了。

软件工程从初期的独立主义到现在，已经发展成一个高度工程化的产业链形式。为了更好地交付软件和服务，人们认识到将开发和运营分隔开来的传统做法已不适合当前的软件服务开发需求，所以就要把开发与运营作为一个整体，这也就是所谓的 DevOps 思想。业界也随之有了对 DevOps 系统的需求，即希望有一套平台或软件能够整合开发和运营交付以及之后的管理工作。所以，从 DevOps 来看，云平台都是为了解决开发人员在开发测试过程中快速搭建环境以及后期运营支持过程中恢复升级监控系统所服务的。近年来，随着云计算产业的发展，DevOps 的概念也在业界变得流行。第一代的 DevOps 系统是基于物理主机部署的，主要靠人工协调来进行业务的自动部署，第二代的 DevOps 系统则是基于 IaaS 进行部署的，借助云计算的资源监控特性，能够实现很好的智能感知等，能够快捷地通过迁移方便地解决硬件单点故障问题，自动做到负载均衡。这是目前很多云计算公司都在极力达到的目标，现在很多公司还不能很好地实现第二代 DevOps 系统。然而，随着容器技术的发展，DevOps 系统有了一个新的方向，即基于容器的部署，容器能够实现应用的跨云迁移，使得应用不会被限定在固有的 IaaS 中。例如，搭建了一套集群，混合了 Vmware 和 Xen 技术，两套系统的虚拟机无法实现实时互相迁移，因为底层的系统存储格式是不一样的；而容器则不存

在这种问题,只需要通过启用对应的容器镜像,就可以启动一个新的服务,这样做就无需关心底层的 IaaS 架构了。

另外,随着将 DevOps 实践引入应用程序生命周期管理,应用程序性能管理(APM)方案的出现则逐渐成为企业实现 DevOps 投资回报的重要前提[4]。对于应用程序的管理,在如今云应用程序普及的情况下,如何整合云应用则是 IT 团队所面临的严峻挑战,而容器技术则为其提供了一个比较好的解决方案。所以,从商业效益上来说,容器技术可以认为是云计算和 DevOps 发展的下一个重要方向。目前已有超过 14 000 个应用建立在 Docker 上,eBay 也正在测试建立在 Docker 上的数据中心软件[5]。

Docker 也可以在虚拟机中运行的很好,这可以让它应用在已有的虚拟化框架中,如私有云和公有云。同样也有可能在容器中运行虚拟机,这有点像谷歌在其云平台中使用容器的方式。只要 IaaS 得到广泛应用,并可按需提供虚拟机服务,那么就有理由期待容器和虚拟机的应用可以并存。还有一种可能,即将容器管理和虚拟化技术进行融合以提供一种两全其美的方法[6]。因为容器能够很好地满足各类 PaaS 功能,所以形象的说,可以把 IaaS 领域的市场看作是苹果的应用程序商店,现在可以添加类似 PaaS 功能,这些功能可以"加载"到采用按需付费的订阅服务模式的 IaaS 产品上[7]。所以随着 IaaS 将各项类 PaaS 的服务添加到自己的服务组合中,PaaS 和 IaaS 之间的界限会变得更为模糊,很有可能 IaaS 和 PaaS 最终会合成一体。

可以汇总出一套通用型解决模式,其中 Docker 分别充当以下几种角色:

(1)Docker 提供经过认证的软件包,并保证其能够与稳定不变的现有基础设施模型顺利协作;

(2)Docker 为微服务 POD 提供出色的容器化运行环境;

(3)在 IaaS 之上使用 Docker,并将其作用于裸机环境等同的运行平台。

是否使用云基础设施仅仅是种自由选项而非强制要求。举例来说,如果出于 DevOps 的目的而考虑建立一套小型自动化开发与测试环境,那么在裸机环境上直接使用 Docker 机制更适合[8]。

2　虚拟机与容器的比较

虚拟主机技术能够让一台物理主机运行多个操作系统,物理主机系统称为 Host OS,而在虚拟机中安装的操作系统则称为 Guest OS。每个 Guest OS 都有自己的计算、存储和网络组件,这些可以通过硬件虚拟化,或者 Host OS 通过把 Guest OS 的指令翻译成物理机指令来实现。所以从理论上来说,Guest OS 可以是任何一款操作系统,与底层的 Host OS 无关。

容器则是一款轻量级的操作系统,它运行在物理机的系统上,直接使用物理机的 CPU 指令,而不需要像虚拟机那样要对指令进行翻译。所以容器不需要像虚拟机那样有太多的开销,并能提供很好的隔离性。虽然虚拟机利用了 RAM 的过度承诺技术(RAM over commitment),容器也表现出比虚拟机更低的系统负载,所以同样的应用,在容器中相比在虚拟机中,性能通常会相当或者更好。经过实际测试,在 Docker 启动时不加载额外软件的话,即只使用基本的系统镜像,所占用的内存仅有几兆,甚至更少。因为 Docker 对于内存消耗主要是在所加载的程序上的,而不像虚拟机需要额外提供给一些系统级的服务,所以对于一台服务器,Docker 能比虚拟机提供更多隔离的容器给用户使用,这样大大提高了物理主机的使用效率。

表 1 比较了虚拟机与容器在性能、隔离性、安全性、网络和存储上的不同之处[9]。

表 1　虚拟机和容器在性能上的比较

参数	虚拟机	容器
客户机系统	每个虚拟机都会有自己的虚拟硬件,都会在自己的内存中加载内核	所有的客户机系统共享系统和内核,内核镜像是加载在物理内存中的
通信	通过网卡设备	使用标准 IPC 机制,例如信号、管道、套接字等
安全性	取决于虚拟机的安全性	使用的是强制的分层访问控制
性能	取决于虚拟机指令转换成物理机指令的性能	几乎是提供与底层物理机一样的性能
隔离性	虚拟机之间共享库、文件等,但不与物理机共享	都能互相访问
开机时间	每个虚拟机启动都需要几分钟的时间	可在几秒内完成启动
存储	需要更多的存储,要存放系统	由于操作系统是共享的,所以需要的存储很少

同时,由于容器也可以安装在 IaaS 所分配的虚拟机上,相比现有的 PaaS,容器技术也更具灵活性。因为现有的 PaaS,大部分都是厂商驱动的,例如 Cloud Foundry(原先由 Vmware 开发)、OpenShift(红帽)等,这些厂商驱动的 PaaS 让广大开发人员和用户被某家厂商牢牢锁定,如果要将应用程序从厂商驱动的平台传送到另一个平台,将会非常麻烦。而容器则不会出现这种问题,因为容器底层就完全可以看作是一个操作系统,就如同一个虚拟机。也正因为如此,现在业界

很看好把容器构建成一个分布式服务。目前比较成熟的就是 Google 的 Kubernetes，这是一款开源的容器管理系统，Google 最新的 Container Engine 就是基于 Kubernetes 设计的。

3 Docker 的读写性能

由于 Docker 本身也是一种虚拟化技术，在使用 Docker 作为 Hadoop 集群部署的环境时，需要保证其在读写性能上不低于现有的虚拟机技术。因为 Hadoop 集群需要存储处理大数据，所以对于 I/O 的读写性能要求很高。为了验证容器与 KVM 的性能，使用了一个测试案例，实验环境为：

硬件环境：4 G 内存，双核 CPU，200 G 磁盘；

软件环境：sysbench 0.4.12，文件块为 16 k。

sysbench 是一款开源软件，主要用于多线程性能测试。

因为实验环境为 Linux 系统，Linux 在实现文件系统时采用了两层结构：第一层是虚拟文件系统，它把各种实际文件系统的公共结构抽象出来，建立统一的以 inode 为中心的组织结构，为实际文件系统提供兼容性[10]。所以，虽然 Docker 使用了自己的一套文件系统，不像 KVM 的虚拟机可以保证使用标准的 Linux 文件系统，但对于实验的影响不大。同时，因为是随机读写的文件，所以为了取得更好的处理效果，把/sys/block/sda/queue/scheduler 的值设为 deadline。该值是经过实验验证后确认的，将所产生的数据分别在 deadline、anticipator、noop 和 cfq 的调度算法下使用 time 命令进行了测试。结果显示，deadline 是耗时最少的。所以为了提高效率，将实验环境都同样设定为相同的调度算法。

随机读写 150 G 的文件，将结果用 gnuplot 绘图，结果如图 1～3 所示。

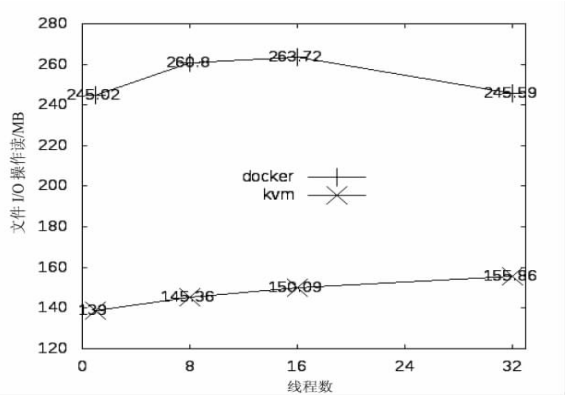

图 1 Docker 与 KVM 在文件读时的性能比较

从实验结果可以看出，使用 Docker 技术，在 I/O 的读写性能上都是优于 KVM 技术的，容器也表现出比虚拟机更低的系统负载。所以同样的应用，在容器中相比在虚拟机中，性能通常会相当或者更好。国外也有一些机构已经开始把 Docker 用于构建大数据处理平台了，例如罗马尼亚的 Cluj－Napoca。Cluj－Napoca 大学就把处理地球观测数据的平台构建在 Docker 上[11]，获得了较好的性能表现。所以说，Docker 技术具有很好的 I/O 性能，能够作为大数据处理工具平台。

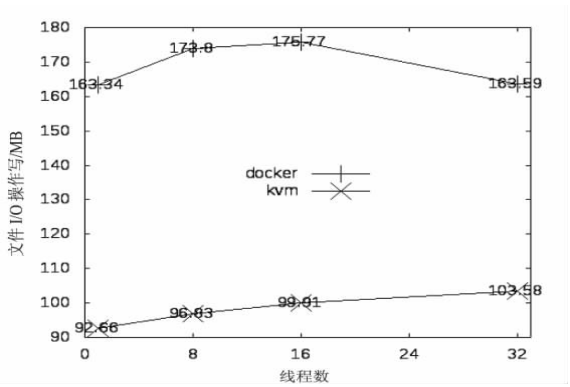

图 2 Docker 与 KVM 在文件写时的性能比较

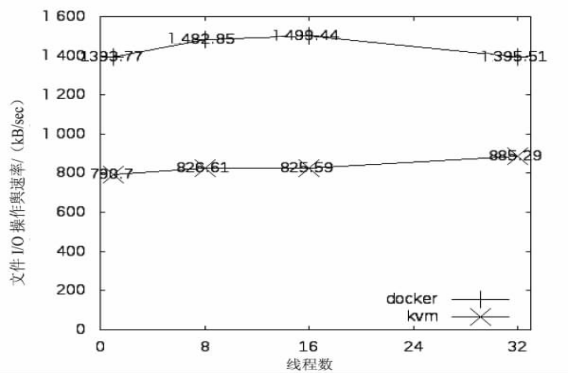

图 3 Docker 与 KVM 在读写文件时的传输速率比较

4 在 Docker 上构建 Hadoop 平台

作为一种特殊的镜像软件，要制作特定的软件镜像，Docker 有其自己的安装打包方式，主要有[12]：

（1）直接在基础镜像上安装软件，然后使用 Docker 命令将其封装成一个新的镜像；

（2）使用 Dockerfile 文件，拉取进出系统镜像后，让镜像根据 Dockerfile 文件的内容自己编译安装。

第 1 种方法的优点是所有操作与真实操作一台虚拟机一样，无需重新学习新内容；缺点是在部署这些镜像时，可能会由于所处环境的不同而造成需要重新修改部分内容，而且，下载完整的镜像所需要消耗的时间较多。

第 2 种方法则只需要在部署时下载该 Dockerfile 文件，然后让系统自己去拉取数据，这样能够减少所需下载的内容，且由于镜像完全是按照 Dockerfile 文件的内容来制作的，所以能够减少人为的干预，从而减少出错；但其缺点是需要花费时间去学习 Dockerfile 文件的机制。

文中使用第 2 种方法来部署 Hadoop，使用的 Hadoop 版本是 2.5.2。

主要完成如下几步：

(1)下载基础系统镜像；

(2)使用 Dockerfile 的内建指令下载安装软件；

(3)使用 Dockerfile 内建指令加载对应的配置文件。

同时，由于 Docker 技术本身还在发展过程中，要使用 Docker 创建能够被外网访问的容器，需要进行一些额外设置。利用容器的底层实现原理，通过脚本动态地为每个容器创建对应的桥接网口。这样能够实现跨服务器间的容器间的互相访问，也更接近真实的应用环境。

通过使用 Dockerfile 文件，完成构建基于 Docker 的 Hadoop 环境。经过测试，Hadoop 能够正常使用。把其与安装在物理主机上的 Hadoop 进行比较，通过 time 命令测试两者运行时间，结果显示两者耗时相差不大，部分情况下 Docker 下的耗时才略高一些。虽然部署的 Docker 数量不多，但能够看出在 Docker 上部署的 Hadoop 在数据读写上的表现非常优异。

5　结束语

Hadoop 作为需要大量读写数据的云计算平台，部署在 Docker 下比部署在传统的虚拟机上有更好的性能表现。文中验证了在 Docker 上读写数据的性能比 KVM 的更高，同时，在 Docker 上部署 Hadoop 具有接近物理主机的资源利用率。并且，由于 Docker 把数据与运行环境进行了分离，所以可以把构建好的 Hadoop 平台作为镜像发布，方便随时添加或替换 Hadoop 节点，这能够提高部署 Hadoop 时的工作效率。Docker 作为容器技术的最佳实践，其性能优势使得其能够替换现有的虚拟化技术。现在国内已经有结合 Docker 与 OpenStack 的研究(如文献[13])和基于 Docker 的 PaaS 平台研究(如文献[14])，这将使得 IaaS 与 PaaS 的界限变得更加模糊，Docker 势必会引领下一场云计算技术的浪潮。

参考文献：

[1] 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，2014.

[2] 肖德时. 深入浅出 Docker[EB/OL]. [2015－01－05]. http://www.infoq.com/cn/articles/docker－core－technology－preview？utm_source＝infoq&utm_medium＝related_content_link&utm_campaign＝relatedContent_articles_clk.

[3] Compton D. Why Docker and CoreOS' split was predictable [EB/OL]. [2015－01－05]. http://danielcompton.net/2014/12/02/modular－integrated－docker－coreos.

[4] Lowy G. Application performance management enables DevOps ROI[EB/OL]. [2015－01－05]. http://www.apmdigest.com/application－performance－mangent－apm－devops－roi.

[5] Garber L. News briefs[J]. IEEE Security and Privacy，2011，9(6)：9－11.

[6] Swan C. Docker：present and future[EB/OL]. [2015－01－05]. http://www.infoq.com articles/docker－future.

[7] Kavis M. Blurring the line between PaaS and IaaS[EB/OL]. [2015－01－05]. http://www.forbes.com/sites/mikekavis/2014/06/02/blurring－the－line－between－paas－and－iaas/.

[8] Shalom N. Do I need OpenStack if I use Docker[EB/OL]. [2015－01－05]. http://pensource.com/business/14/11/do－i－need－openstack－if－i－use－docker.

[9] Dua R，Raja A R，Kakadia D. Virtualization vs containerization to support PaaS[C]//Proc of IEEE international conference on cloud engineering. Boston，MA：IEEE，2014：610－614.

[10] 黄　刚，徐小龙，段卫华. 操作系统教程[M]. 北京：人民邮电出版社，2009.

[11] Bica M，Bacu V，Mihon D，et al. Architectural solution for virtualized processing of big earth[C]//Proc of IEEE international conference on ICCP. Cluj Napoca：IEEE，2014：399－404.

[12] Turnbull J. The Docker book[M]. [s. l.]：Amazon Digital Services，Inc.，2014.

[13] 张忠琳，黄炳良. 基于 OpenStack 云平台的 Docker 应用[J]. 软件，2014，35(11)：73－76.

[14] 鞠春利，刘印锋. 基于 Docker 的私有 PaaS 系统构建[J]. 轻工科技，2014(10)：80－80.

(上接第 98 页)

[6] 刘兴民，赵连军. 基于 GStreamer 的远程视频监控系统的关键技术研究[J]. 计算机应用与软件，2011，28(5)：243－246.

[7] 徐　家，陈　奇. 基于 V4L2 的视频设备驱动开发[J]. 计算机工程与设计，2010，31(16)：3569－3572.

[8] 刘登诚，沈苏彬，李　莉. 基于 V4L2 的视频驱动程序设计与实现[J]. 微计算机信息，2011，27(10)：56－58.

[9] 刘喜龙，石中锁. 基于 H264 的嵌入式视频服务器的设计[J]. 微计算机信息，2005，21(1)：133－134.

[10] 李世平. H.264 三大开源编码器之评测报告[R/OL]. 2005. http://blog.csdn.net/sunshine1314/archive/2005/06/19/397895.

[11] 刘　浩，胡　栋. 基于 RTP/RTCP 协议的 IP 视频系统设计与实现[J]. 计算机应用研究，2002，19(10)：140－143.

[12] 刘尚麟，刘　军. GStreamer RTP 插件的改进及应用[J]. 信息安全与通信保密，2009(1)：91－92.

[13] 孙彦景，李世银，董　杨. 基于 RTP 的嵌入式网络化视频采集压缩系统[J]. 计算机工程与设计，2006，27(16)：2939－2942.