应用于高能物理计算的容器技术研究

高能物理研究所计算中心

汇报人：谭宏楠
2017-07-04

目录
CONTENTS

01 背景

高能物理计算集群规模不断扩大、高能物理计算正在向高性能、并行化方向发展。传统集群系统出现资源利用率不高、应用迁移复杂、多应用支持困难等问题。

虚拟机云计算技术的易于整合多种资源，规模弹性可伸缩，运行成本低等特点解决了上述问题，但其存在性能损耗、结构复杂等问题。

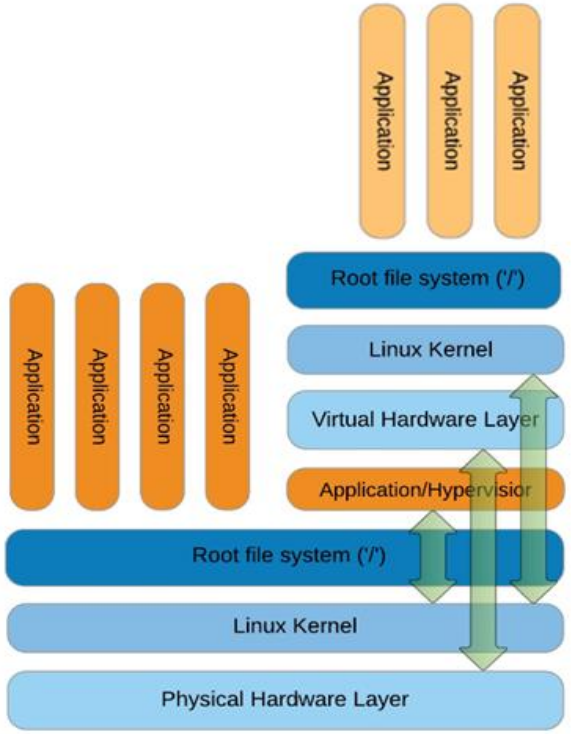
容器级的虚拟化技术，在隔离度、资源消耗、启动速度等方面具有更大的优势，如果将其应用于高能物理计算，可降低虚拟化对数据分析处理带来的损耗，提升计算资源管理的灵活性。

01 背景

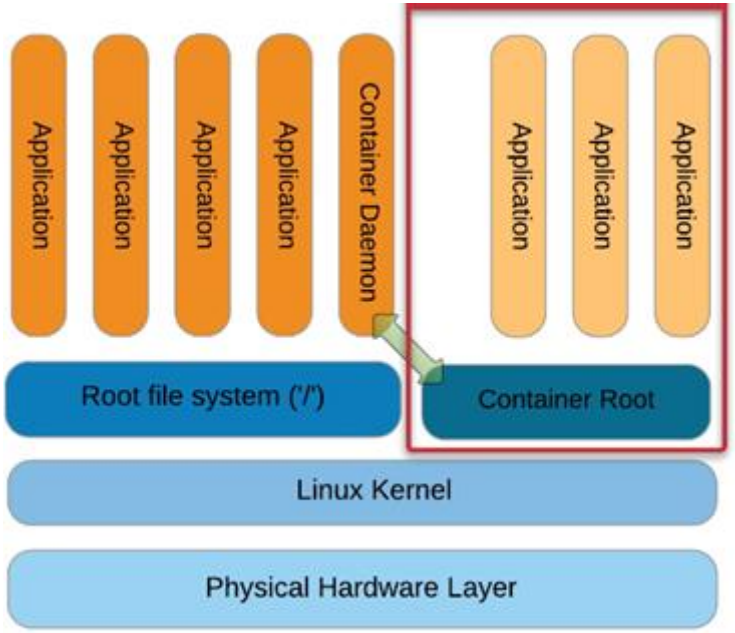
容器技术介绍

– 虚拟机技术利用Hypervisor虚拟化CPU、内存、磁盘等硬件设备，模拟了一套完整的操作系统。而容器技术无需中间的虚拟硬件层，利用容器实现隔离，与宿主机共享同一套操作系统内核。

– 容器技术使用内核的cgroup和namespace等技术实现虚拟操作系统环境，实现了容器的轻量级特性。



虚拟机

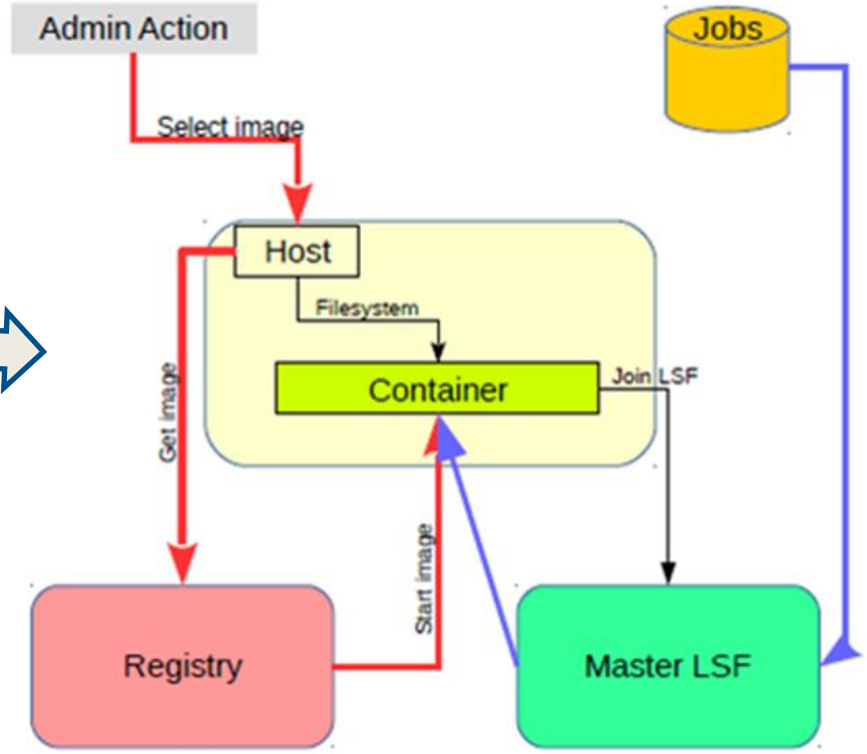


容器

02 应用现状

- Docker是目前较为流行的一种开源的容器引擎，提供了镜像打包和封装的有效方式，引入Docker Registry 对镜像进行管理；
- Docker对于镜像分层的创新设计，使得容器在磁盘占用、性能、效率以及资源消耗等方面相对于传统的虚拟化有了极大的提高和改善。

意大利INFN 已将CMS的网格计算服务迁移到Docker容器集群中。

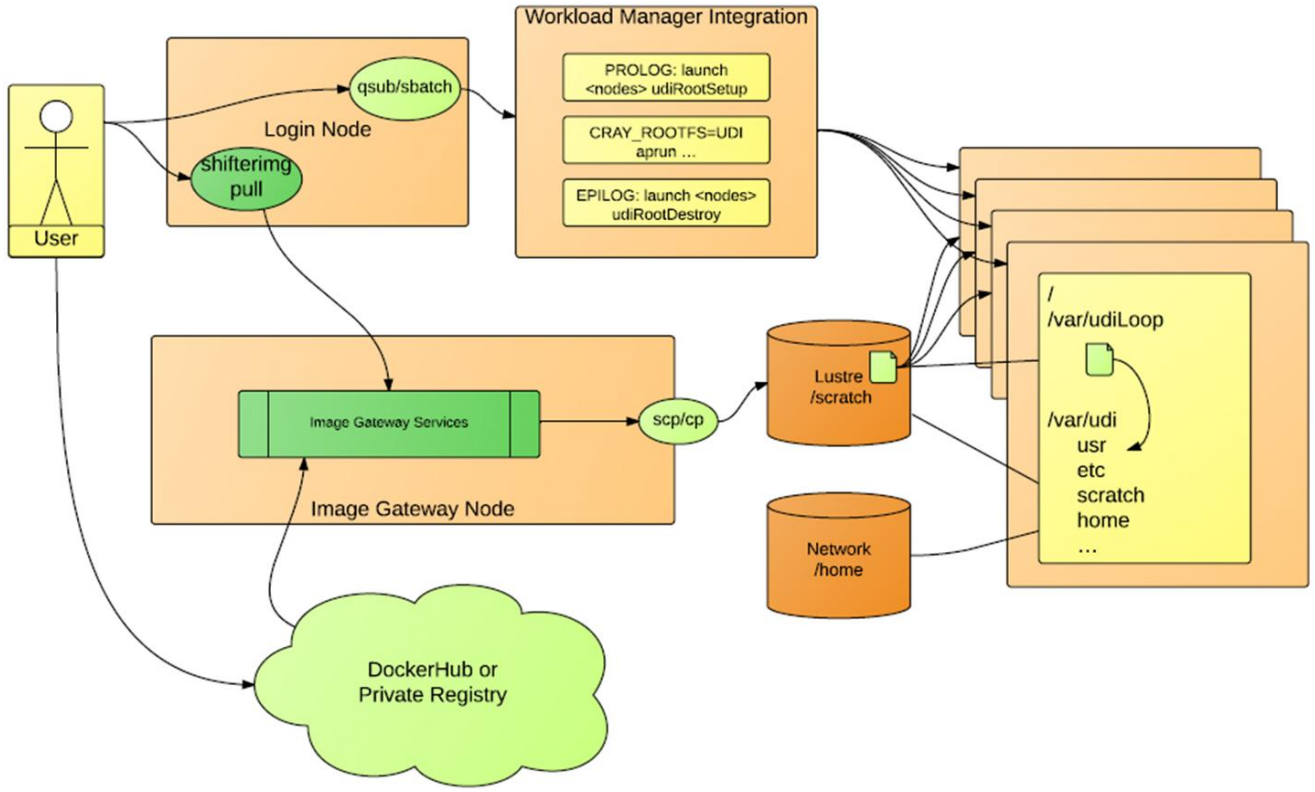


- 通过Docker提供的命令批量启动容器；
- 在容器上运行集群作业调度器LSF，将容器作为 LSF 的普通计算节点使用，接收并运行作业；
- 容器仅作为一个静态的计算节点运行。

02 应用现状

容器在HPC中的应用

- Shifter 由NERSC（美国国家能源研究计算中心）与Cray（克雷）合作开发，是一种将容器技术应用于HPC环境中的解决方案；
- Shifter引用Docker的镜像机制，提出UDI（user-defined，user-provided images）的概念。



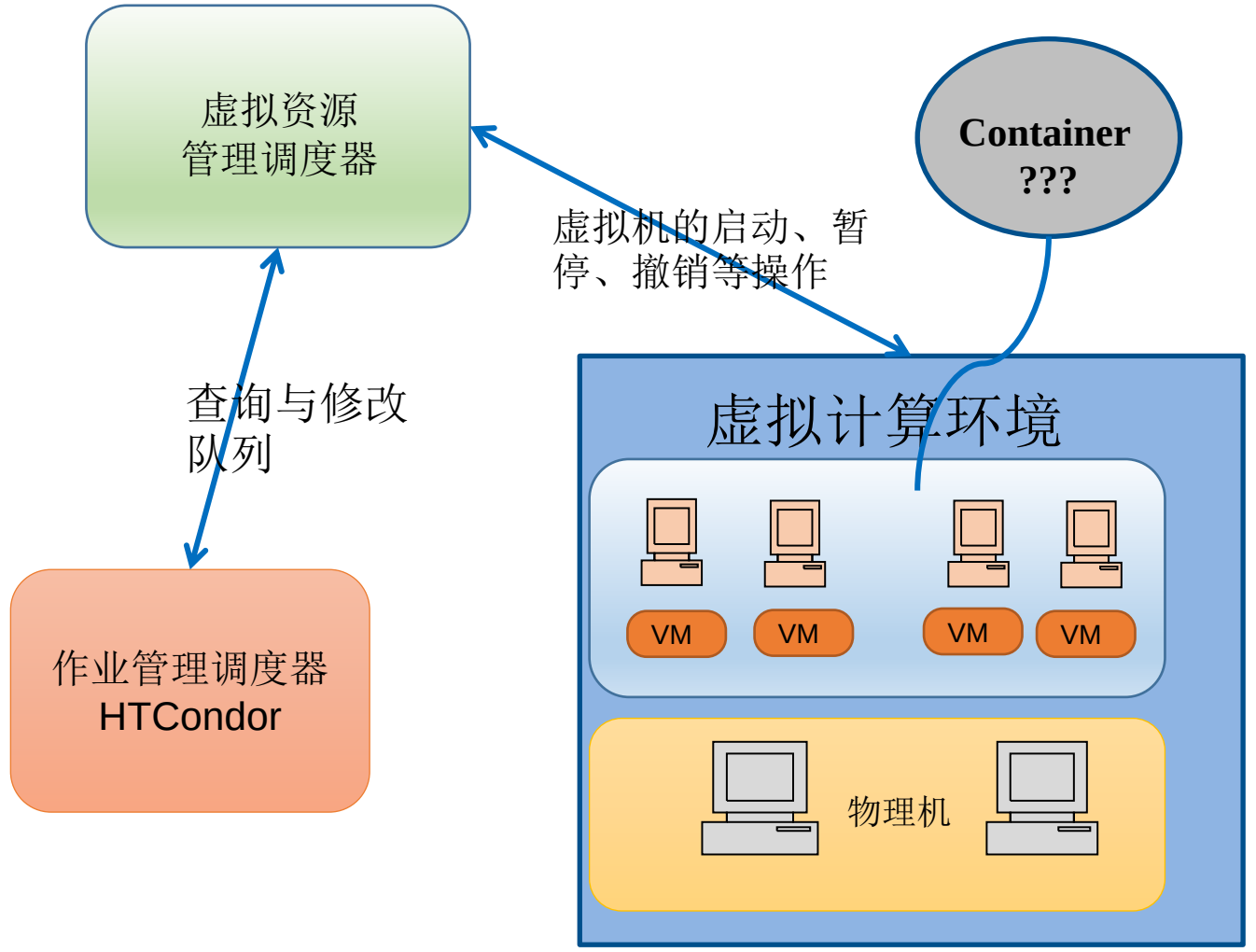
- Shifter由两个组件 Image gateway和UdiRoot组成；
- 制作好的UDI上传至镜像仓库后，通过Image gateway组件获取镜像，之后镜像被上传至共享位置；
- UdiRoot 组件创建loop device，挂载Shifter镜像，并在loop device 上创建chroot 环境，之后便可在该环境下运行程序或应用。

02 应用现状

高能所虚拟计算环境

- 采用KVM 虚拟机屏蔽底层基础设施的异构性，动态调整集群的资源分配；
- 虚拟集群系统对用户透明；
- 在一定程度上提高了系统资源的利用率和作业运行效率；
- 但虚拟机启动速度慢，CPU、I/O性能损耗较大。

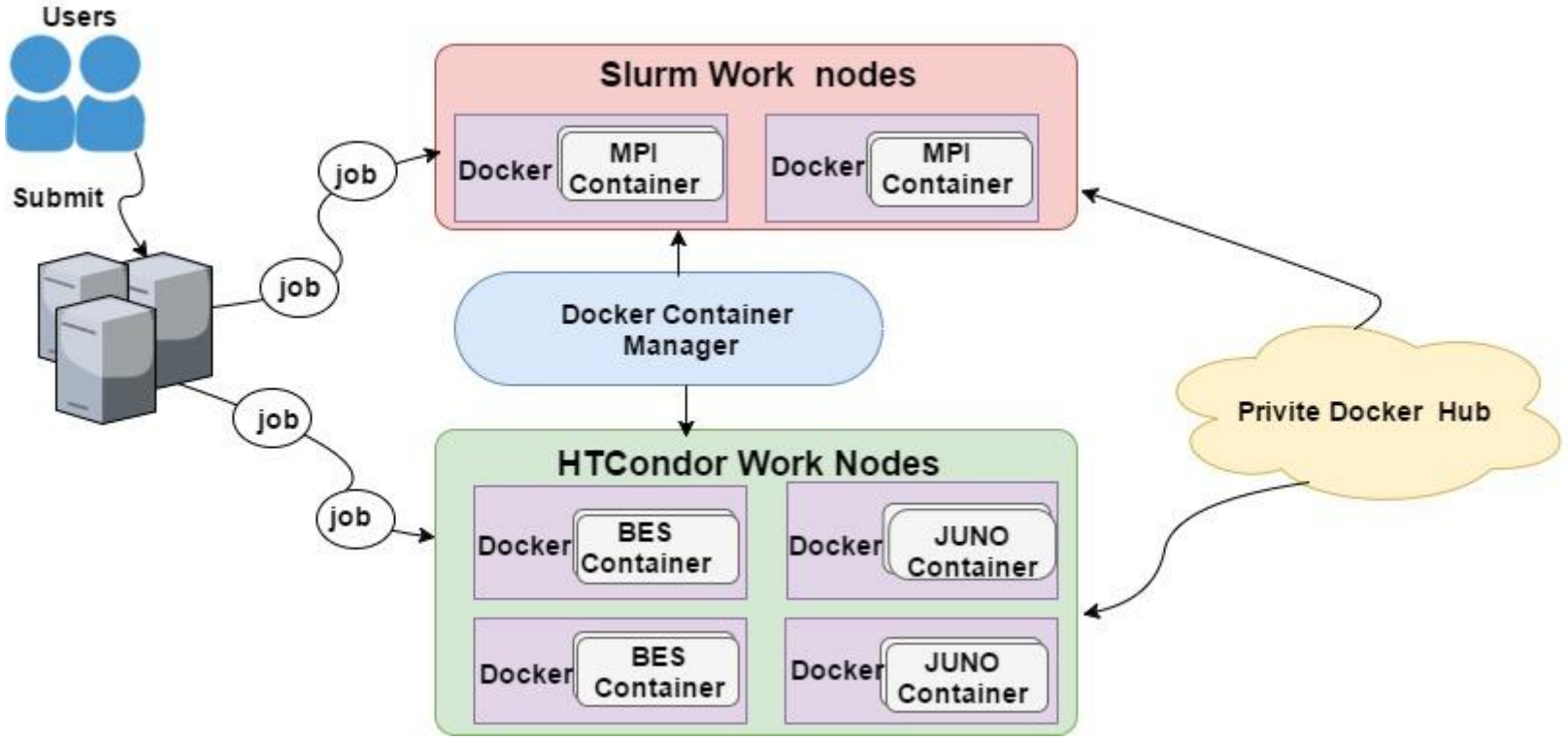
如何将容器加入虚拟计算环境是需要解决的问题



03 方案设计

工作流程

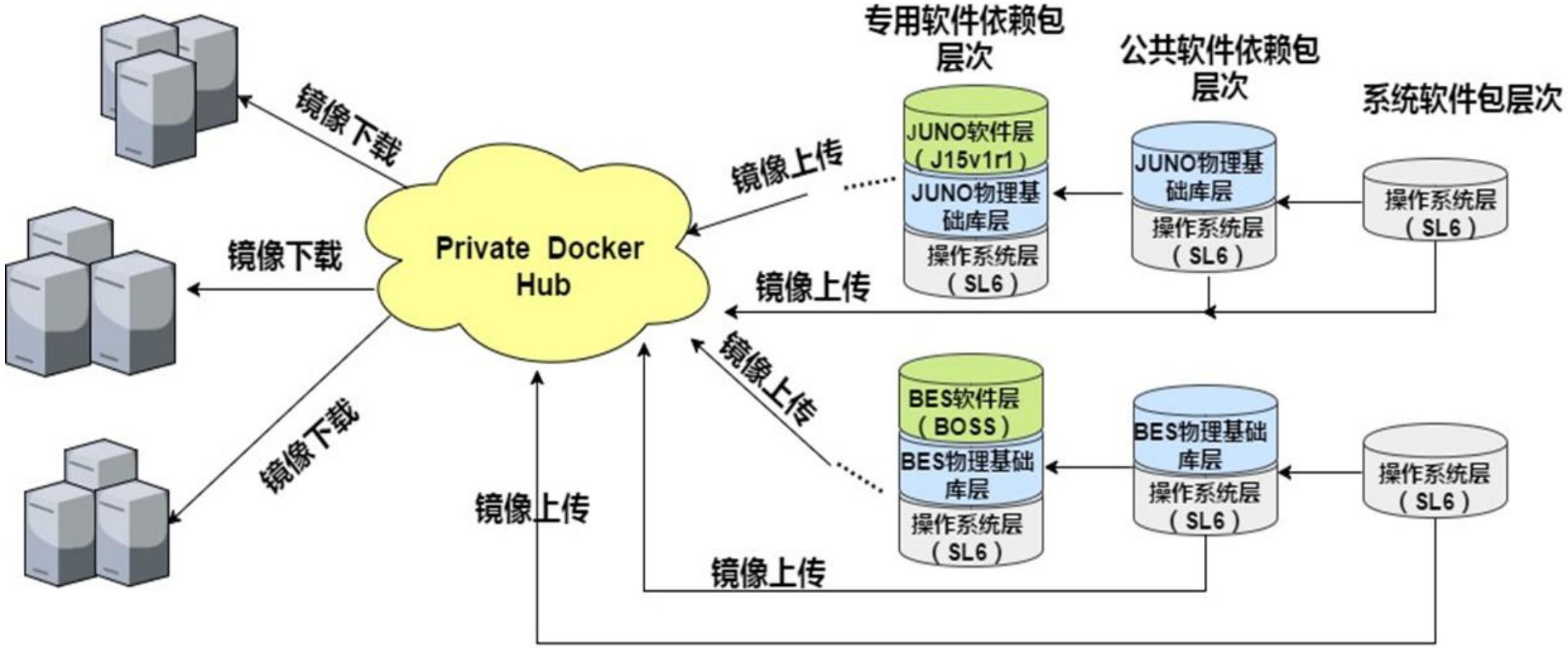
- Privite Docker Hub：专用高能物理镜像仓库。
- 用户正常提交作业，作业调度系统按常规方式接收作业、分配资源并运行作业。
- 作业在容器内运行，完成后容器停止并释放资源。
- Docker Container Manager: 容器资源管理器，批量部署、启动、停止容器。



03 方案设计

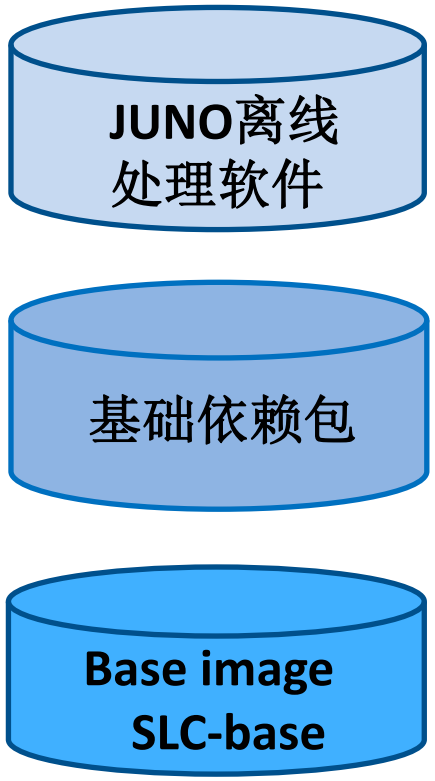
容器镜像定制分发设计

- Docker镜像文件为容器提供运行时所需的程序、库、资源、配置等；
- 利用Docker镜像分层技术，可以为不同高能物理实验灵活定制专用的Docker 镜像文件；
- 按“系统镜像层”、“公共软件镜像层”和“专用软件镜像层”对离线数据处理软件中的程序包和软件库逐一分类标签。



03 方案实现

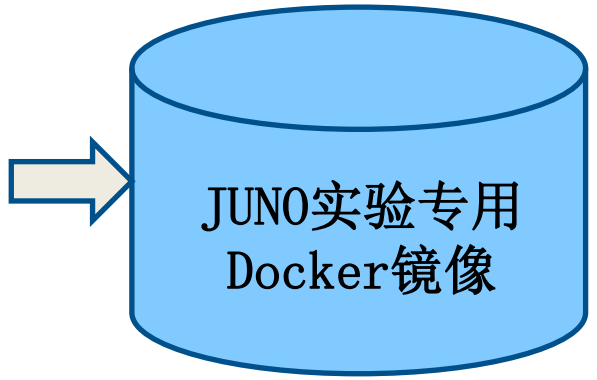
基于JUNO实验的镜像定制




```
From slc-base
MAINTAINER tanhongnan

RUN yum install -y  make gcc-c++ gcc binutil \
    && yum install -y libX11-devel libXpm-devel
libXft-devel libXext-devel \
    && yum install -y install  mesa-libGL-devel  ftgl-
devel mysql-devel\
    && yum install -y fftw-devel  graphviz-devel \
    && yum install -y avahi-compat-libdns_sd-devel
python-devel \
    && yum install -y libxml2-devel gsl-static gsl-
devel \
    && yum install -y qt-devel && yum clean all

CMD /bin/bash
```



- JUNO实验专用镜像文件经优化后大小为600M ；
- 目前用于JUNO 实验的虚拟机镜像大小为5.2G；
- 小尺寸 的镜像文件便于存储、分享和更新。

04 性能测试

测试环境

测试软硬件环境

- 硬件环境：(1)CPU：Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz
  (2)内存：32GB

- 软件环境：(1)操作系统：Scientific Linux 7.2
  (2)Docker版本：Docker version 17.03.1-ce

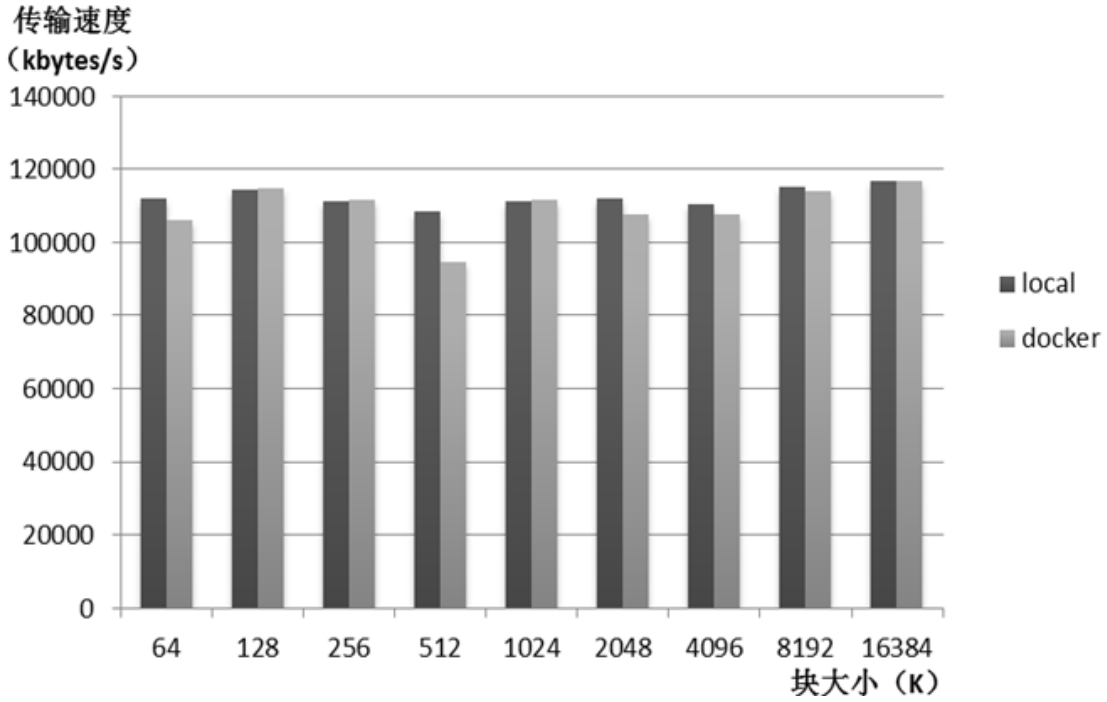
测试平台及工具

- 搭建物理机、虚拟机、容器三种统一测试平台，三种平台均是实际计算环境中的计算节点
- IOzone工具测试磁盘读写性能
- HPESPEC06工具测试CPU性能
- 综合测试

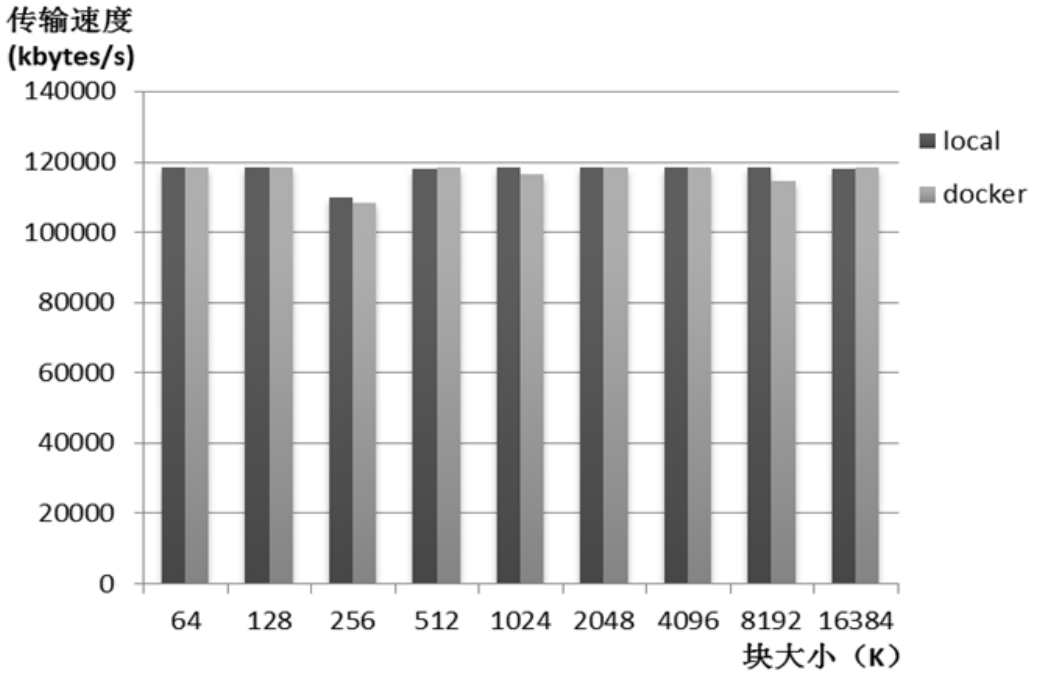
04 性能测试

I/O性能测试

- 高能所计算中心使用分布式文件系统Lustre为JUNO等多个实验提供海量存储服务，管理共享磁盘空间和实验数据；
- JUNO实验的作业运行时会有大量的数据输入及输出，磁盘的读写性能对计算效率有很大影响；
- 选用IOzone对其测试。



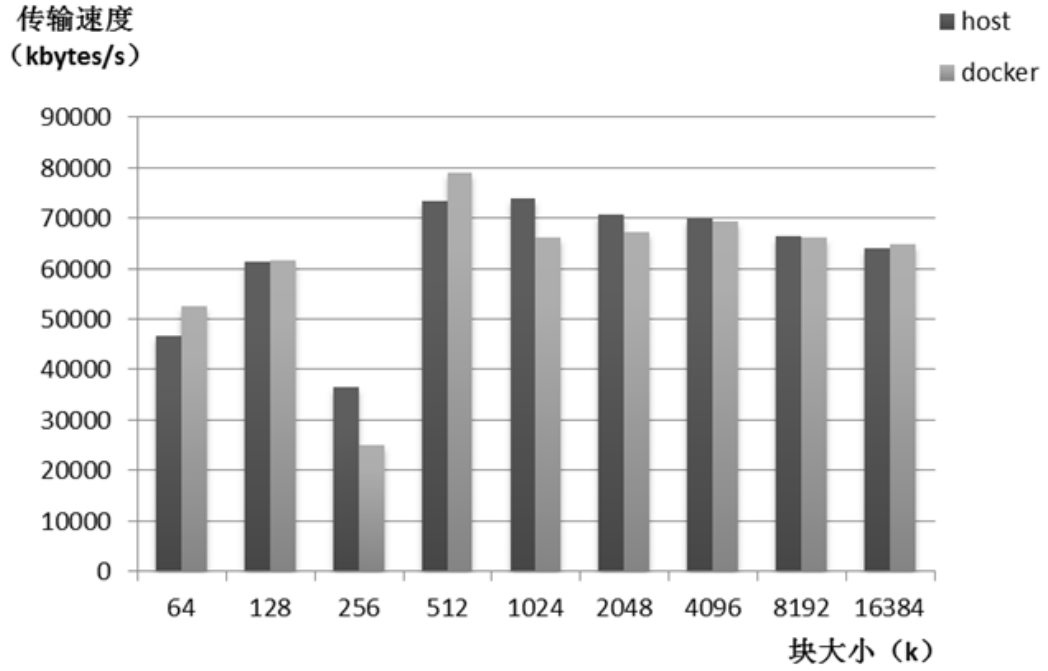
IOzone顺序读测试结果



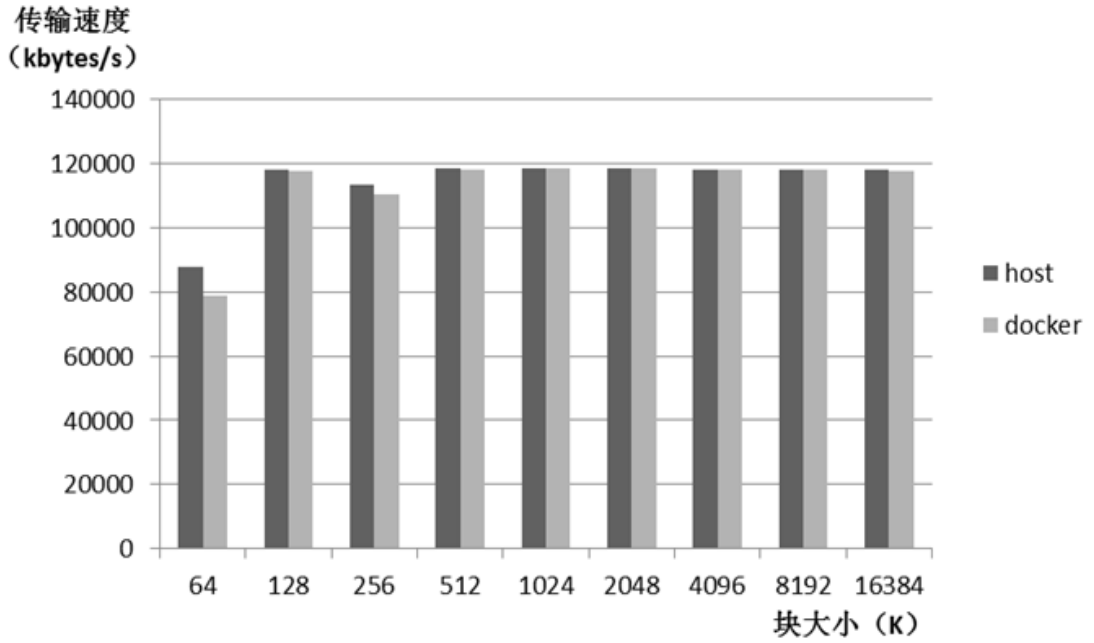
IOzone顺序写测试结果

04 性能测试

I/O 性能测试



IOzone随机读测试结果



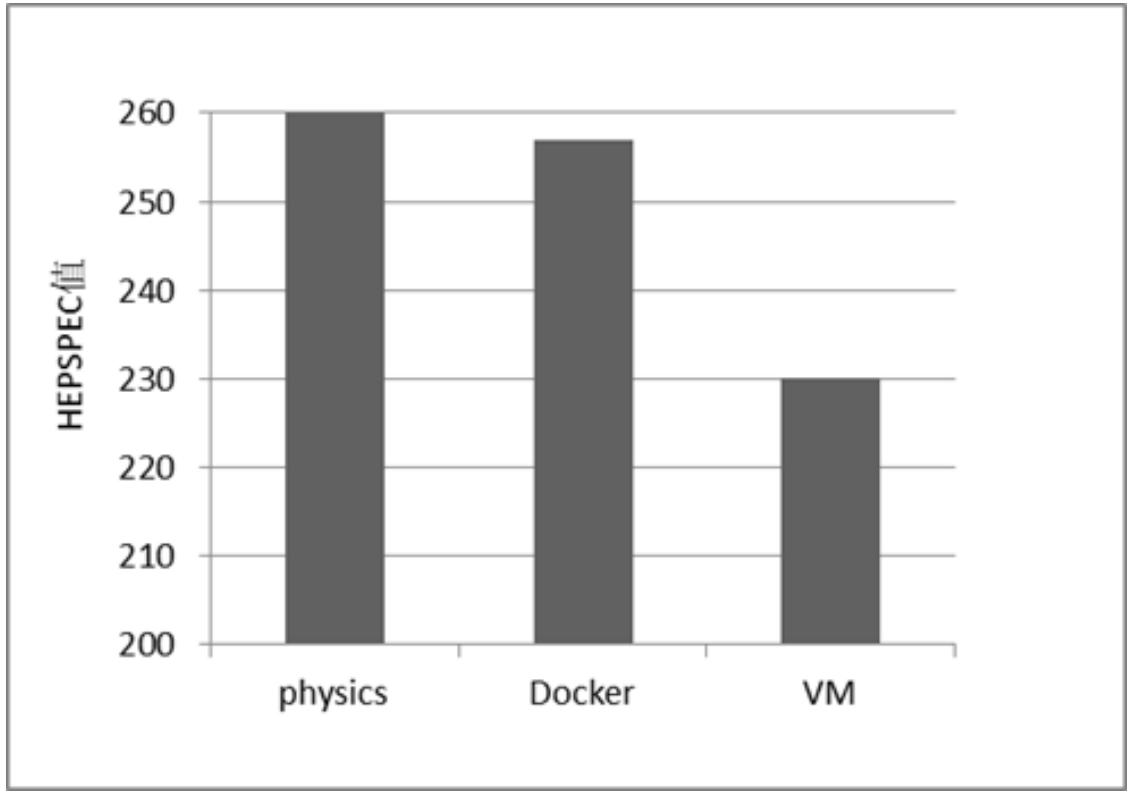
IOzone随机写测试结果

测试结果表明：在顺序读、顺序写，随机读、随机写四个场景中，容器的I/O性能与宿主机的I/O性能相比，损耗很低。

04 性能测试

HEPSPEC06工具测试

- HEPSPEC06 是用于高能物理计算的标准检测程序；
- 运行HPESPEC06得到的测评分数代表机器应用于高能物理计算的能力高低；
- 测试值越大，表示CPU的计算能力越强。

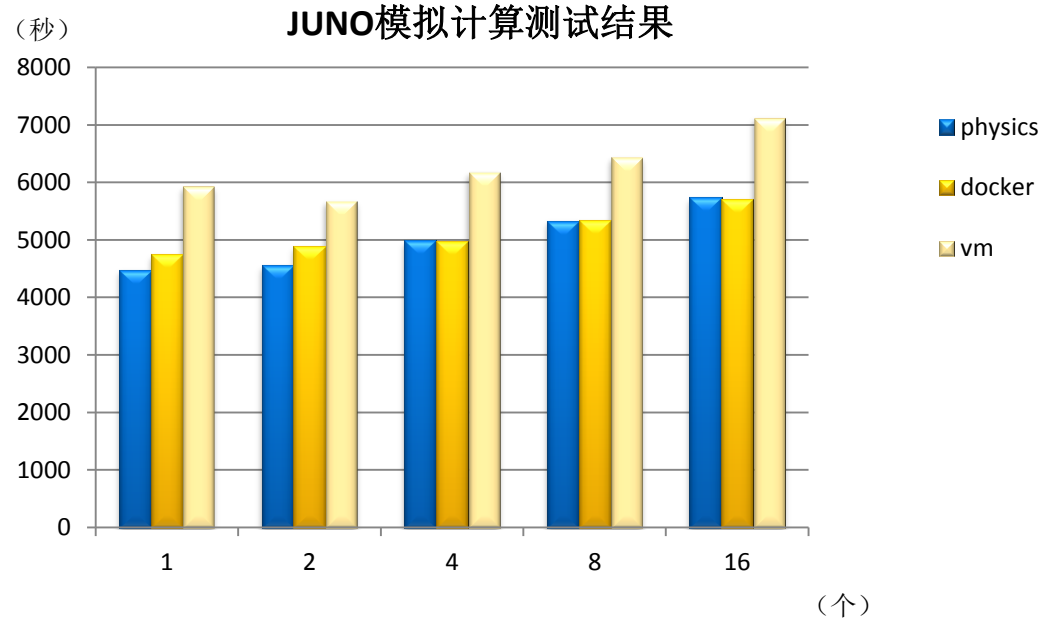


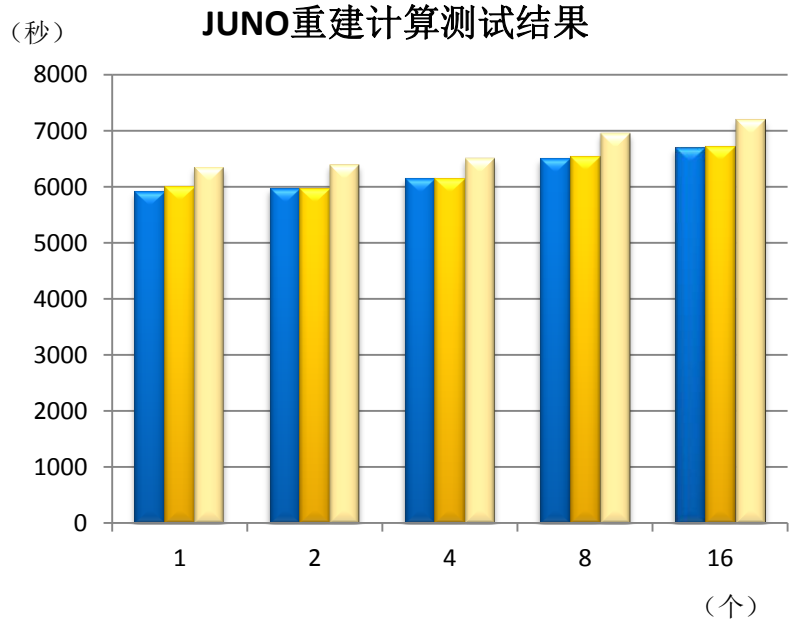
虚拟机的HEPSPEC 分值比物理机低11%；而容器的HEPSPEC值与物理机分值相比，差距<1%。

04 性能测试

综合测试

- 将同样规模的测试计算任务分别提交到物理机、虚拟机以及容器运行；
- 以JUNO 离线软件中的模拟作业和重建作业为样本测试。





容器的性能损耗小于1%（模拟&重建）
虚拟机的性能则损失则高达24.1%（模拟）和7.5%（重建）

05 总结与下一步计划

总结

- 容器共享宿主机Linux 内核，性能接近物理机；
- 针对JUNO实验的镜像定制方案及测试结果初步确定了将容器技术应用于高能物理计算环境的可行性；
- 容器轻量快速的特点更适合于资源快速整合，比虚拟机可更好实现资源池化，保证规模弹性扩展的实时性与可用性。

下一步计划

- 将容器与当前的作业调度系统结合，将计算作业迁至容器中运行，并对容器资源管理系统进行设计、调优。

谢谢大家！