

 **TiD2016**
质量竞争力大会
软/件/研/发/顶/级/盛/会/

SOFTWARE
DEVELOPMENT
**下一代
软件研发**
更高 更新 更深

Docker 引领测试革新

孙远 华为中央软件研究院



我的经历

美国风河系统公司

- 负责linux build system、analysis tools、workbench测试工作。

华为中央软件研究院





- 带领团队进行容器OS、docker测试工作，并参与项目过程改进。
- 参与开源社区：补充ltp社区user namespace特性测试用例和docker社区中的测试用例。
- 完成《docker进阶与实战》测试章节的编写工作。



内容

- 测试技术面临的挑战
- Docker与虚拟机
- Docker组件、生态
- Docker开源社区的测试
- Docker技术在测试中的应用
- Devops与容器软件供应链
- Docker与自动化测试
- Docker测试实践(编译测试、外围包测试、测试私有云)

计算机技术的革新

- 1980 单机系统、无网络-挑战：运行速度 
- 1990 网络系统-挑战：组网、快速发布 
- 2000 分布式、虚拟化-挑战：系统扩容性 
- 2010 容器微服务系统-挑战：业务快速发布且运行不间断 
- 20xx 未来技术-挑战：综合挑战

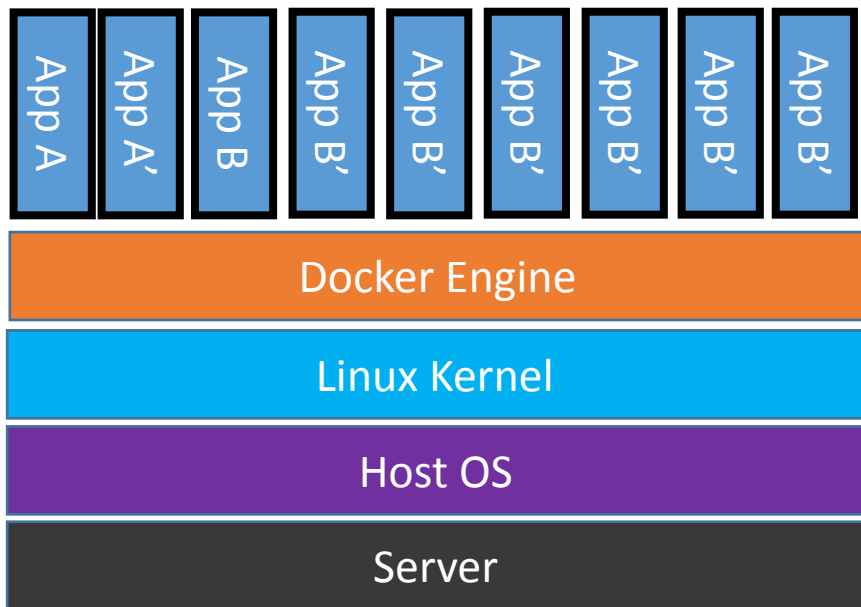
传统软件开发流程的痛点

- 开发、测试、运维环境不统一；无法准确获取客户的软件环境。
- 开发在提交代码前未做充分的测试。
- 开发无法复现测试报出的bug，开发与测试之间相互推诿。
- 配置测试环境的时间较长，测试自动化成本高。

测试的挑战

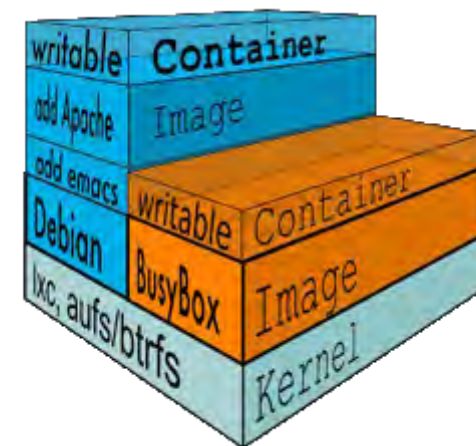
- 如何配置一致的测试环境
- 快速部署软件
- 并行执行测试
- 成功的复现bug
- 创建清洁的测试环境
- 测试工具的正确配置
- 快速的部署多个测试主机
- 快速导入测试数据
- 快速清理测试环境
- 快速保留、复制、恢复测试环境

Docker架构

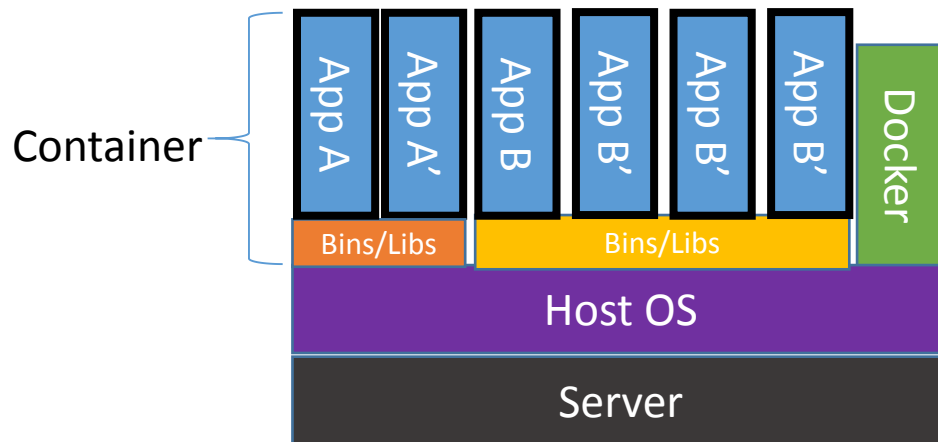
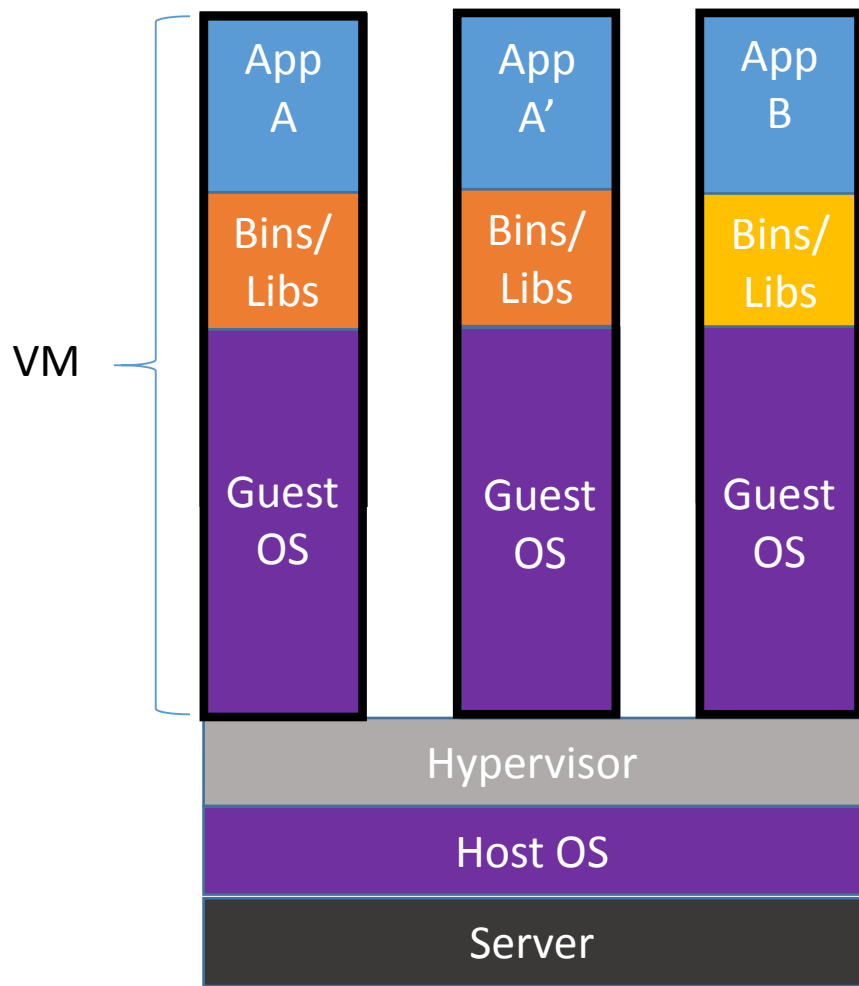


支撑docker的核心技术

- Cgroups
资源限制
- Namespace
资源隔离
- Union File System
Docker镜像的基石、分层存储



VM vs Docker



指标项:

- 启动时间
- 部署时间
- 系统资源占用
- 环境分享
- 环境释放时间

- 模拟不同arch环境
- 扩展性
- 应用可移植性
- 系统可靠性
- 系统安全性

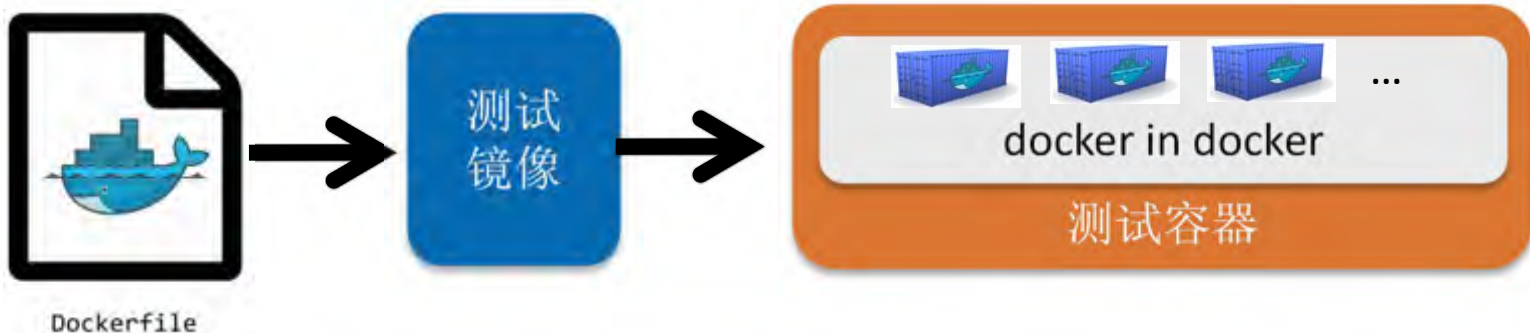
Docker组件与生态



- Docker Machine
- Docker Compose
- Docker Swarm
- Docker Notary
- Docker Plugins
- Runc
- Registry
- 容器编排
- 容器网络
- 容器存储方案
- 轻量级容器OS
- 容器安全
- 镜像仓库
- Docker store

Docker 开源社区的测试

```
$ git clone https://github.com/docker/docker.git  
$ cd docker && make test
```

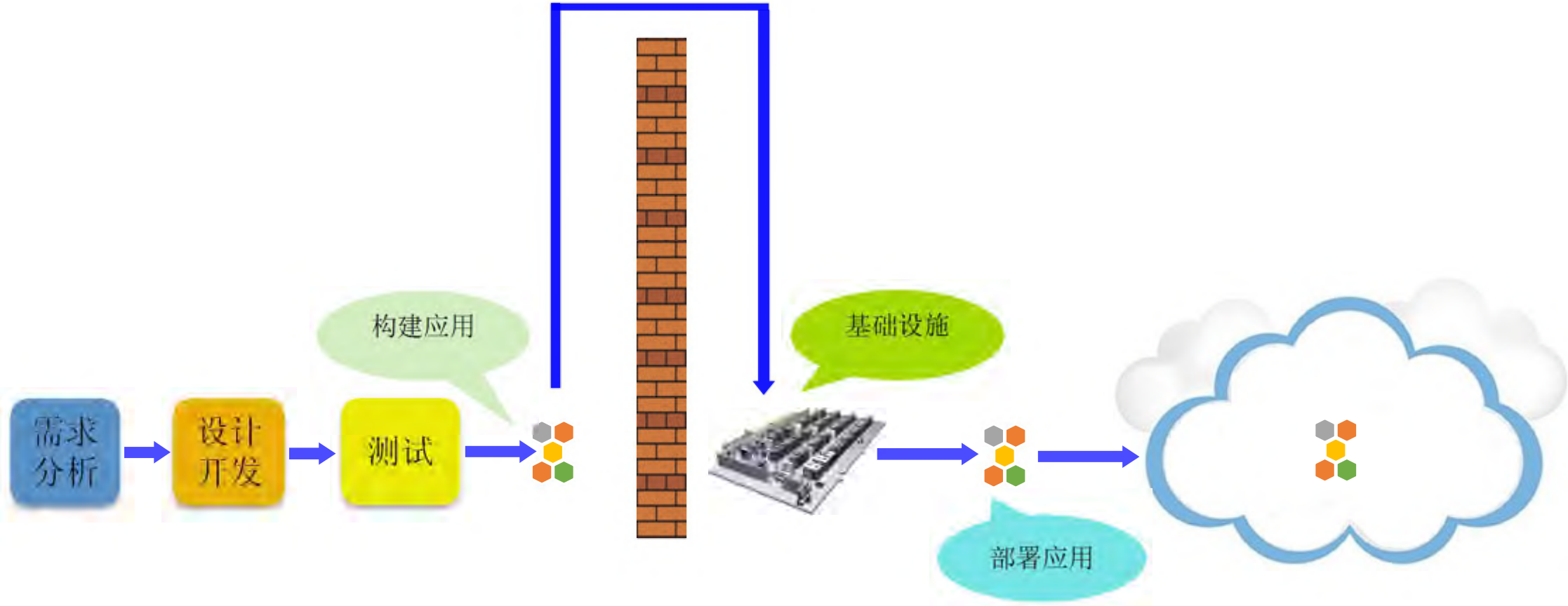


```
PASS: docker_cli_pull_test.go:106: DockerHubPullSuite.TestPullAllTagsFromCentralRegistry  
PASS: docker_cli_pull_test.go:137: DockerHubPullSuite.TestPullClientDisconnect  
PASS: docker_cli_pull_test.go:20: DockerHubPullSuite.TestPullFromCentralRegistry  
PASS: docker_cli_pull_test.go:69: DockerHubPullSuite.TestPullFromCentralRegistryImplicitRefParts  
PASS: docker_cli_pull_test.go:46: DockerHubPullSuite.TestPullNonExistingImage  
PASS: docker_cli_pull_test.go:96: DockerHubPullSuite.TestPullScratchNotAllowed  
OK: 991 passed, 38 skipped  
PASS
```

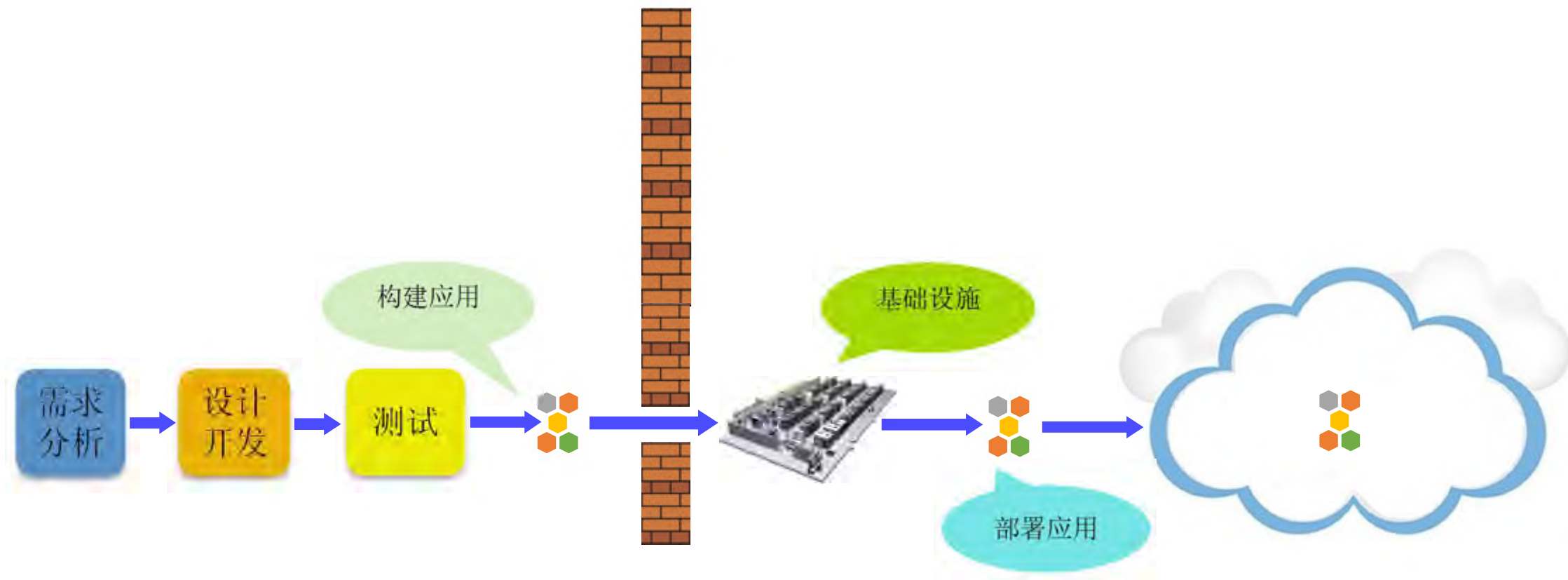
Docker对测试的革命性影响

- 让单元测试运行得更顺畅。-快速构建环境,测试驱动开发
- 让虚拟机不再困扰集成测试和功能测试。 -避免多任务共享虚拟机
- 让测试团队和客户丢掉冗长的配置文档。
- 可以轻松地复现客户报告的bug。
- 通过Dockerfile可以梳理好测试镜像制作的流程。
- 方便软件厂商将成熟的测试套或测试工具通过镜像共享。
- 利用Docker生态中的工具可以快速创建可伸缩的测试环境，大大减少了测试所消耗的时间。
- 优越的性能指标。-秒级启动、快速挂载、快速清理
- 轻松的恢复测试环境（包括内存）-CRIU技术 Checkpoint Restore In Userspace

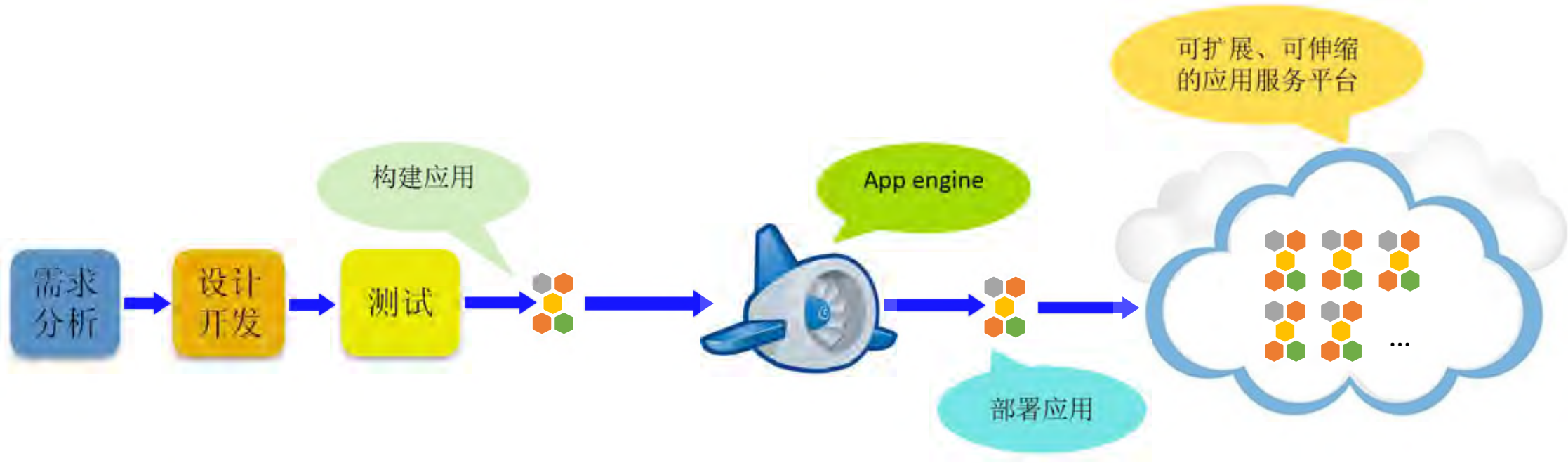
DevOps出现之前



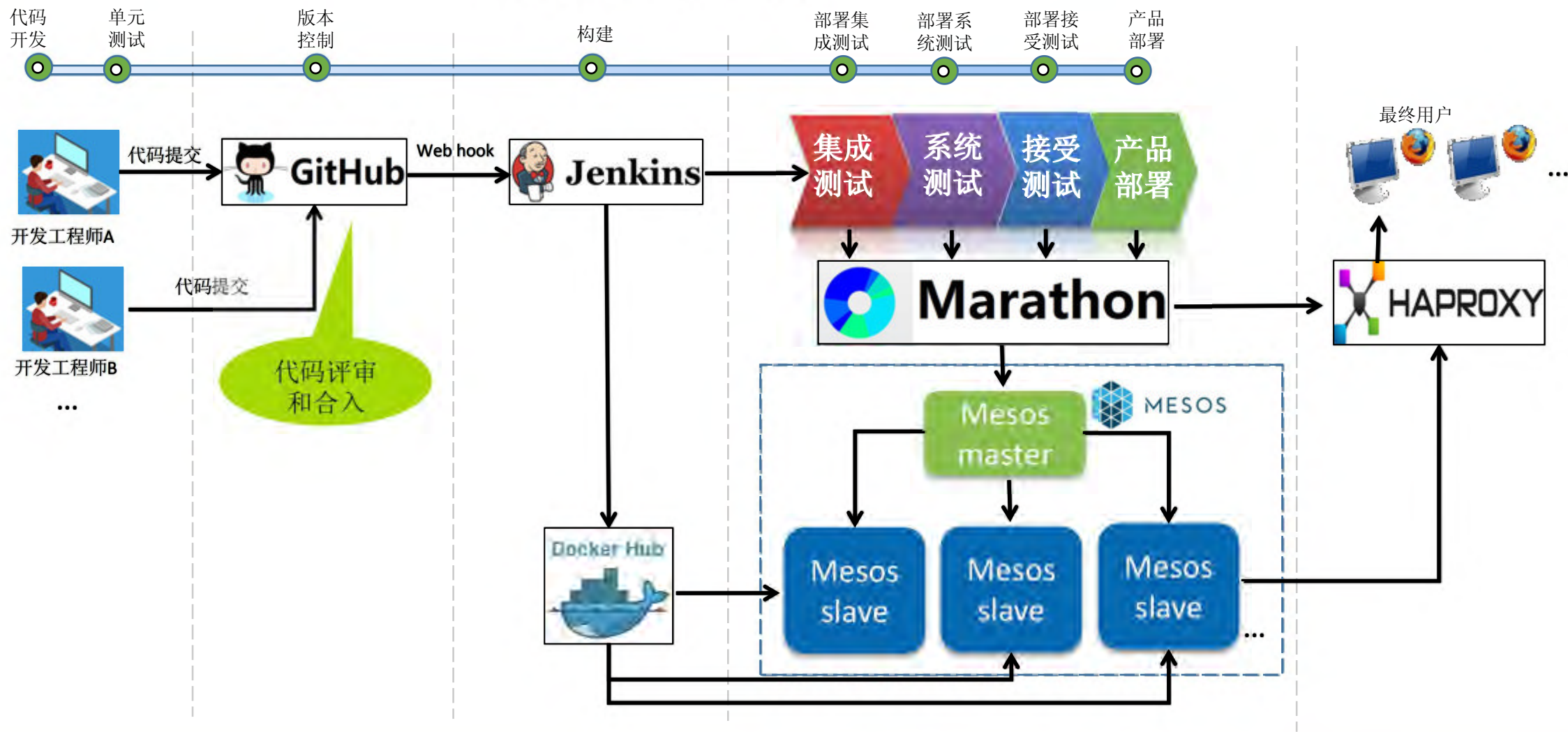
DevOps时代



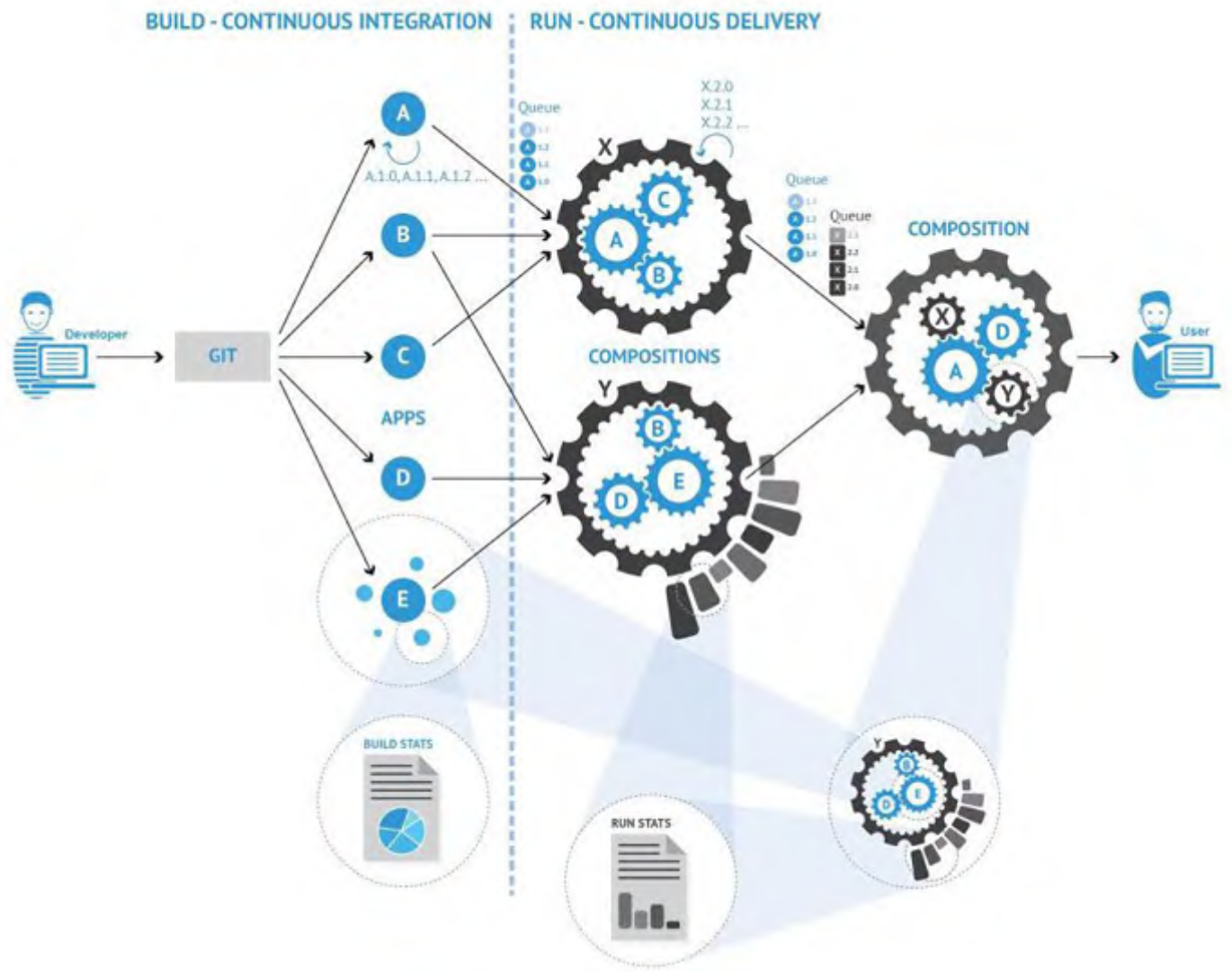
后DevOps时代



测试部署



容器软件供应链



Docker与自动化测试

- 什么样的软件测试适合自动化?
- 是否需要将现有的测试通过docker进行自动化?
- 原先自动化测试运行在虚拟机中, 是否需要将其移植到docker中?
- 如何将Jenkins与docker相结合使用?



Docker公司的理想与现实

理想

- Build, Ship, and Run Any App, Anywhere.

现实

- 由于容器与主机共用内核，如果容器需要使用不同的内核版本就不得不更换主机内核。
- 不能修改内核参数或者自主定制内核。
- 对内核版本有依赖性，**Docker**通常需要3.10或以上版本的内核。
- 在容器中加载或卸载内核模块会影响到主机和其他容器。
- 跨主机容器间通信能力不足。
- 无法像qemu一样模拟嵌入式系统运行环境。

思考

- 编译系统测试
- 数据库测试
- 与内核相关的网络测试
- 内核测试套LTP
- Web应用测试
- 应用软件安装测试
- ARM嵌入式软件模拟测试

哪些测试场景
适用于docker?



容器化编译系统测试

测试场景

- 安装不同版本的编译软件: gcc3.4, gcc4.3 ...,gdb...
- 配置不同的产品依赖软件包: glibc,java,...
- 在不同发行版中进行源码编译。


软件
产品




物理机

虚拟机

容器



ubuntu 14.04, 14.10,
15.04,15.10



red hat 7.0, 7.x
8.0, 9.0



openSUSE 11.x

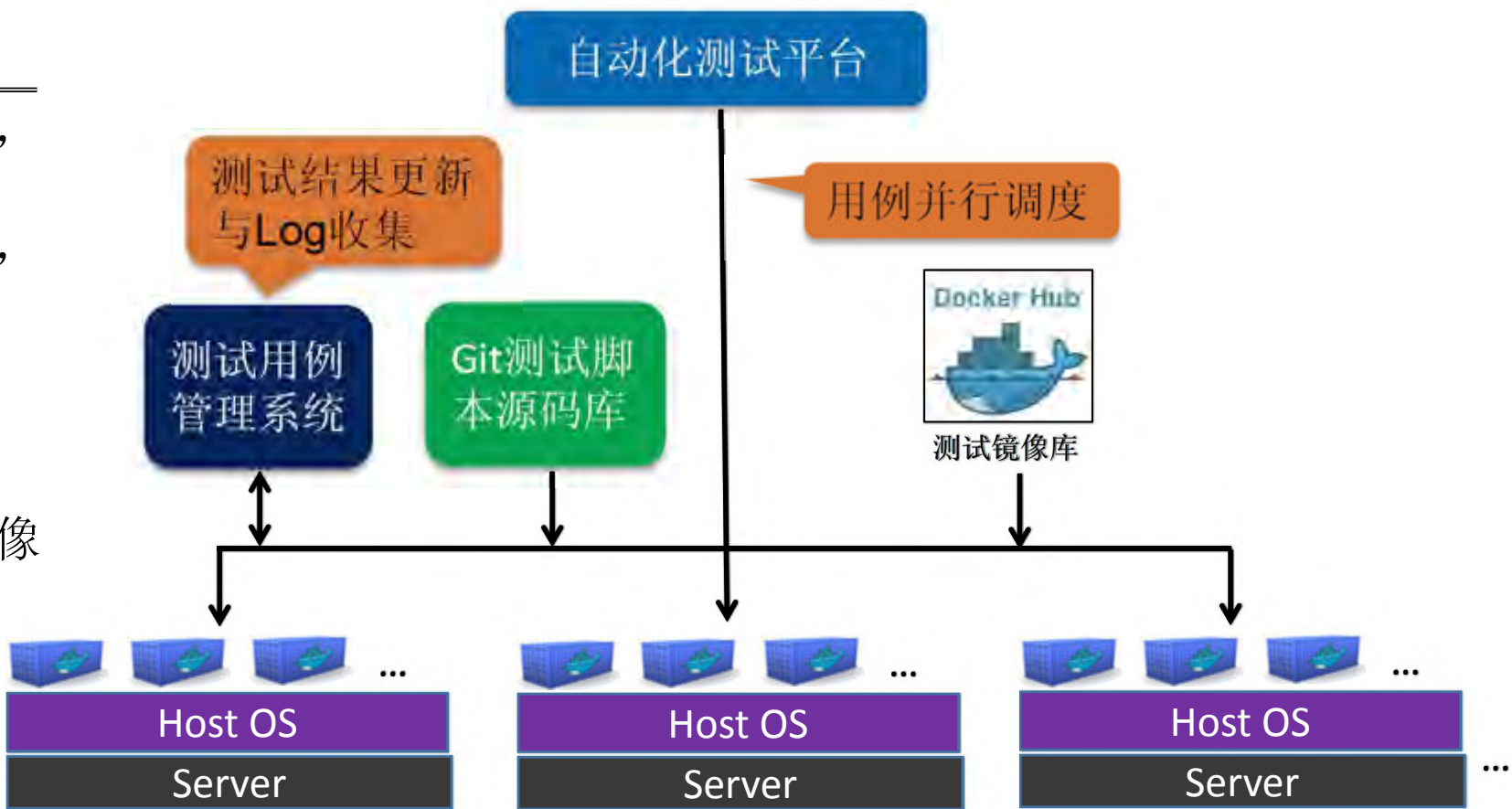


fedora 23, 24

容器化编译系统测试

优化手段

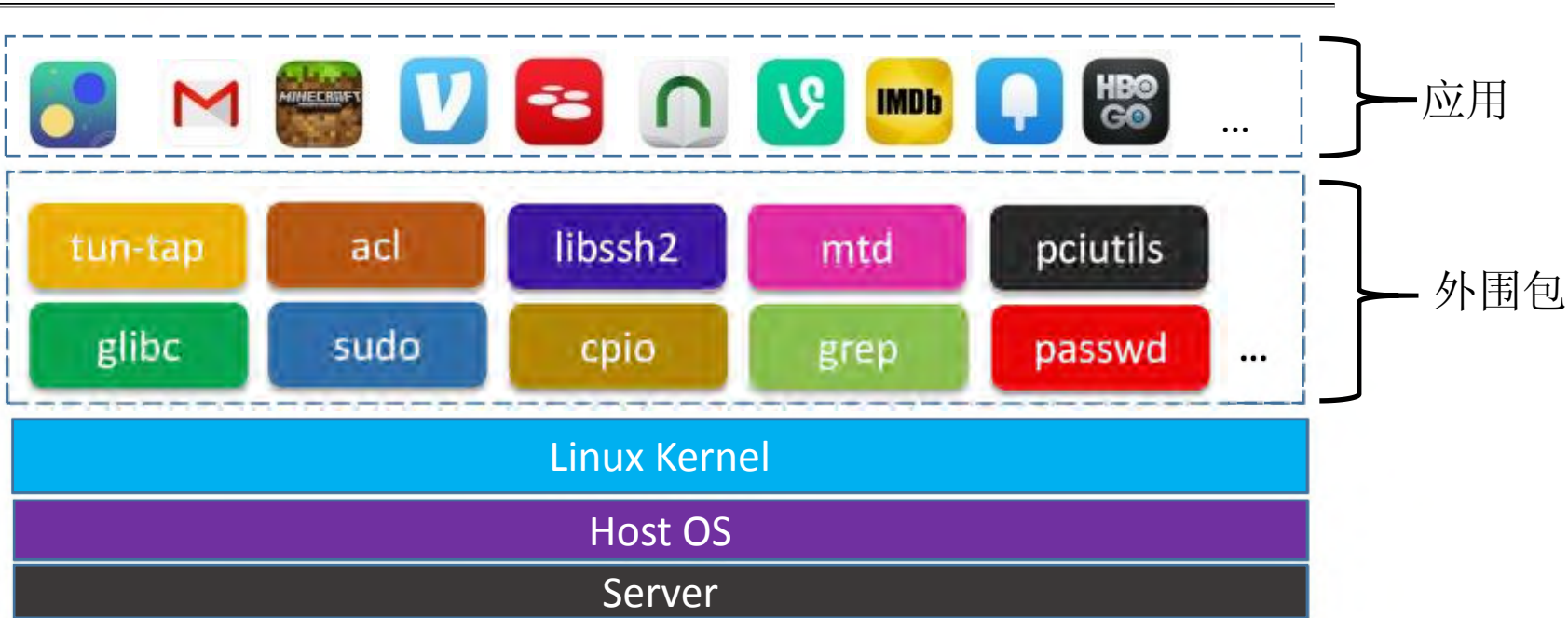
- 部署效率：容器间共享测试套，秒级部署。2分钟->10秒
- 环境使用率：容器替代虚拟机，减少资源损耗
效率提升1倍
- 细化大粒度用例
- 用例脚本并发执行
- 使用dockerfile快速生成测试镜像



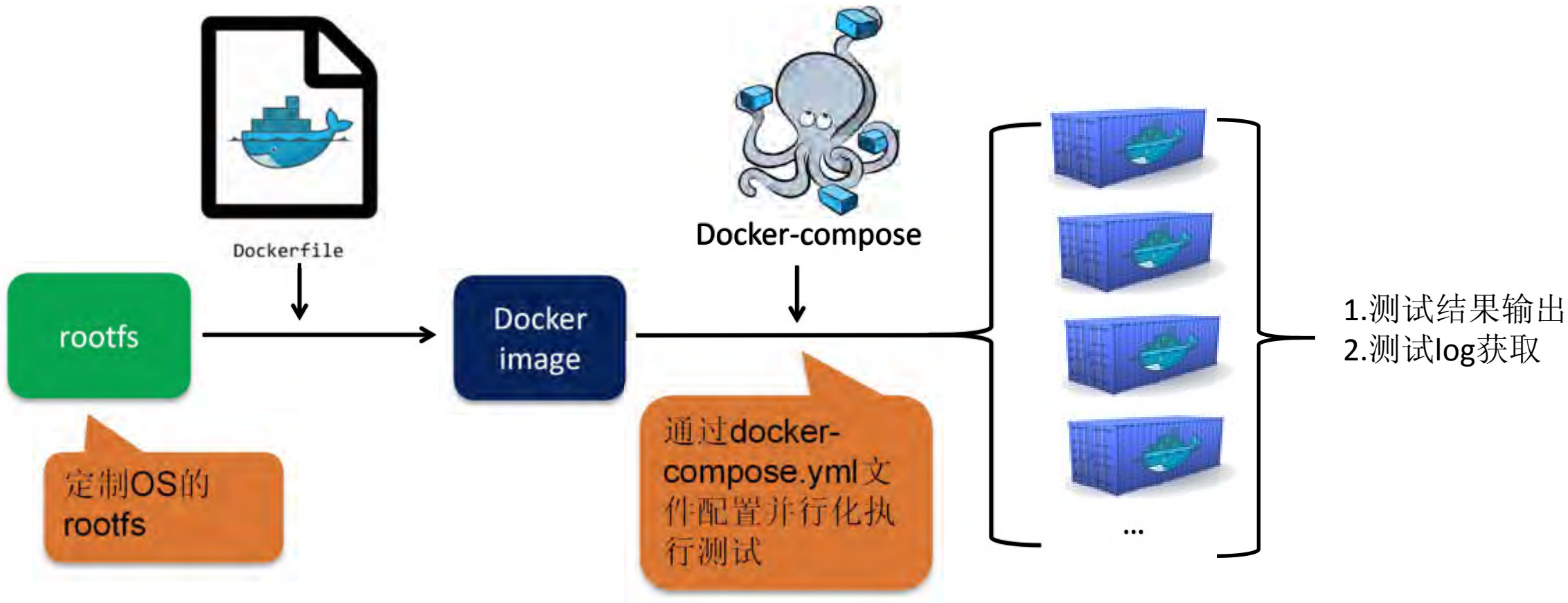
Linux 外围包测试

测试场景

- 在OS启动后执行外围包功能测试(runtime)
- 需要确认外围包功能和依赖
- 覆盖不同类型的Linux OS



Linux外围包测试



Linux外围包测试

传统部署特点:

- 每套环境独占一台主机
- 测试串行执行，不易并发
- 环境释放时清理工作依赖于程序员的技能
- 无法解决多个外围包的环境污染问题
- 外围包编译环境不易统一，导致测试结果不统一
- 测试网络包时需要至少两台主机

Docker部署特点:

- 多套环境可以在同一主机上部署
- 测试并行执行，提高了cpu利用率
- 环境释放时清理工作由docker接管
- 容器可快速启动与关闭，每次都是清洁的环境
- 通过镜像保存编译环境，确保环境统一
- 测试网络包时只需要在一台主机中启动两个容器

思考与demo演示

Docker是通过什么方式来加速测试执行的？



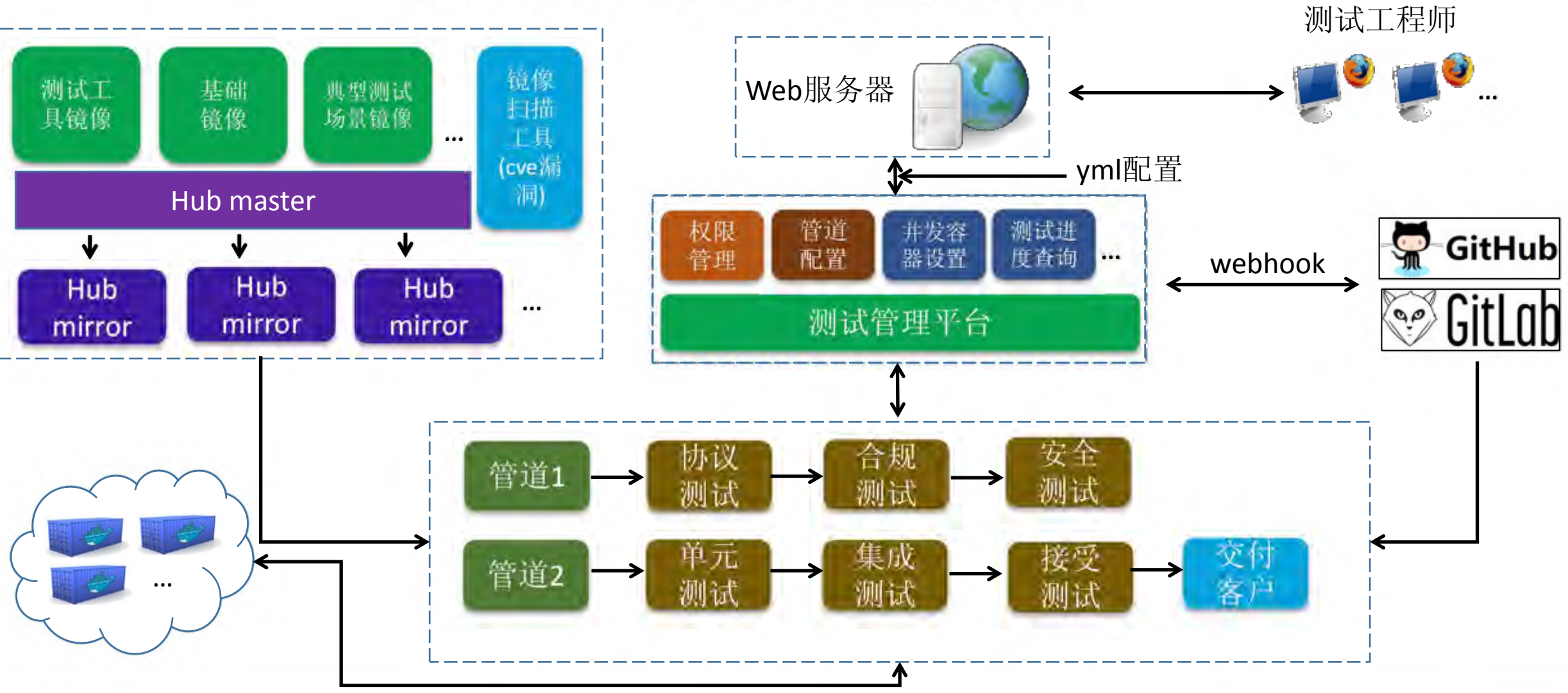
测试私有云解决方案

通过自定义WebUI管理界面+后台管理系统+容器调度+测试专用仓库

底层技术:

- 容器OS微小化
- 磁盘限额
- 容器独立ip
- 资源管理（限定cpu、内存、io等消耗）
- 可弹性伸缩
- 高可用容器引擎
- 测试专用docker仓库（多租户）
- 容器监控
- 容器热迁移
- 容器卷存储
- 容器安全加固

测试私有云解决方案



思考

Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence.

- Edsger W. Dijkstra



引用资料

- 《The Humble Programmer》 Edsger W. Dijkstra
- 《why-docker-website》
- 《docker进阶与实战》
- 《Software Testing》 (2nd Edition) Ron Patton
- 《Testing Cloud Services: How to Test SaaS, PaaS & IaaS (Rocky Nook Computing)》 (1st Edition) Kees Blokland, Jeroen Mengerink, Martin Pol
- 《Using Docker/Software Containers for Automated Testing》 Pini Reznik

提问环节



联系我:

孙远 华为中央软件研究院

研究方向: 容器技术、docker、linux内核、软件测试、自动化测试

Email: sunyuan3@huawei.com,
yuan.sun82@gmail.com

<https://github.com/sunyuan3>