

携程 Docker 实践



- ◆ 容器对携程的价值
- ◆ 携程部署容器的方式和现状
- ◆ 携程部署Docker架构
- ◆ Docker容器运行
- ◆ Docker 监控

提高资源利用效率

更加灵活的应用发布

未来Windows容器的部署

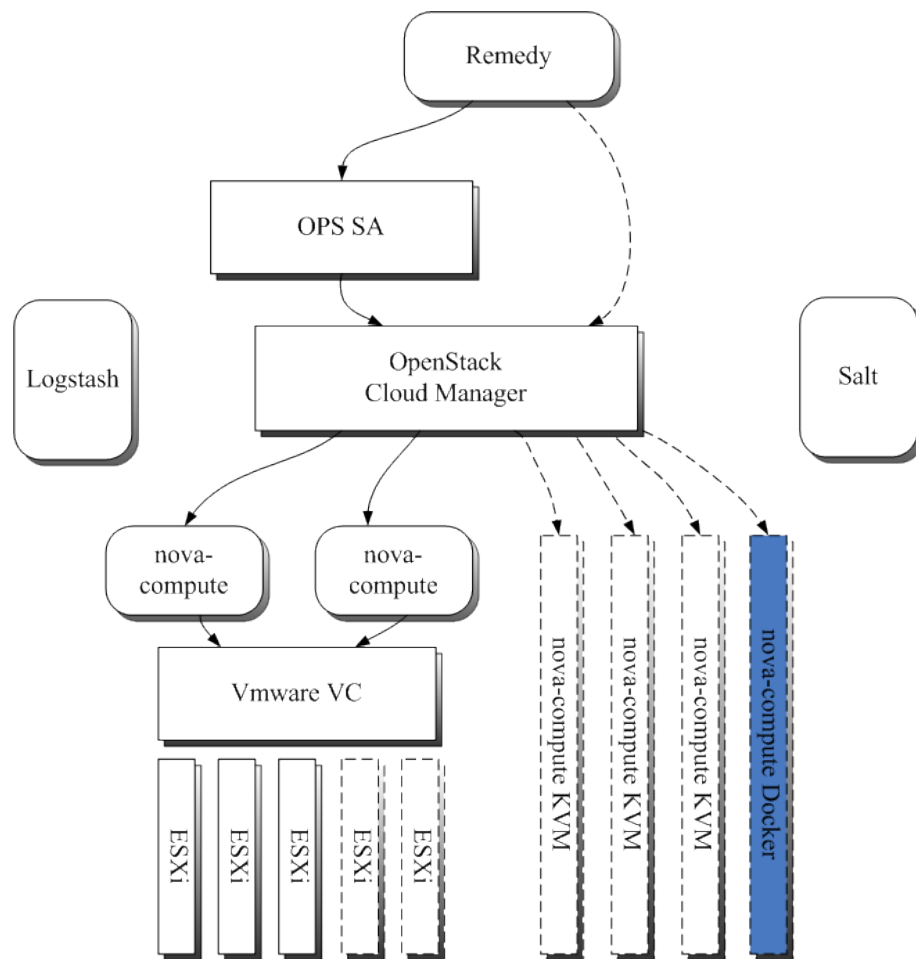
以VM的方式运行容器

- 优点：方便平滑过渡
- 缺点：容器优势体现少

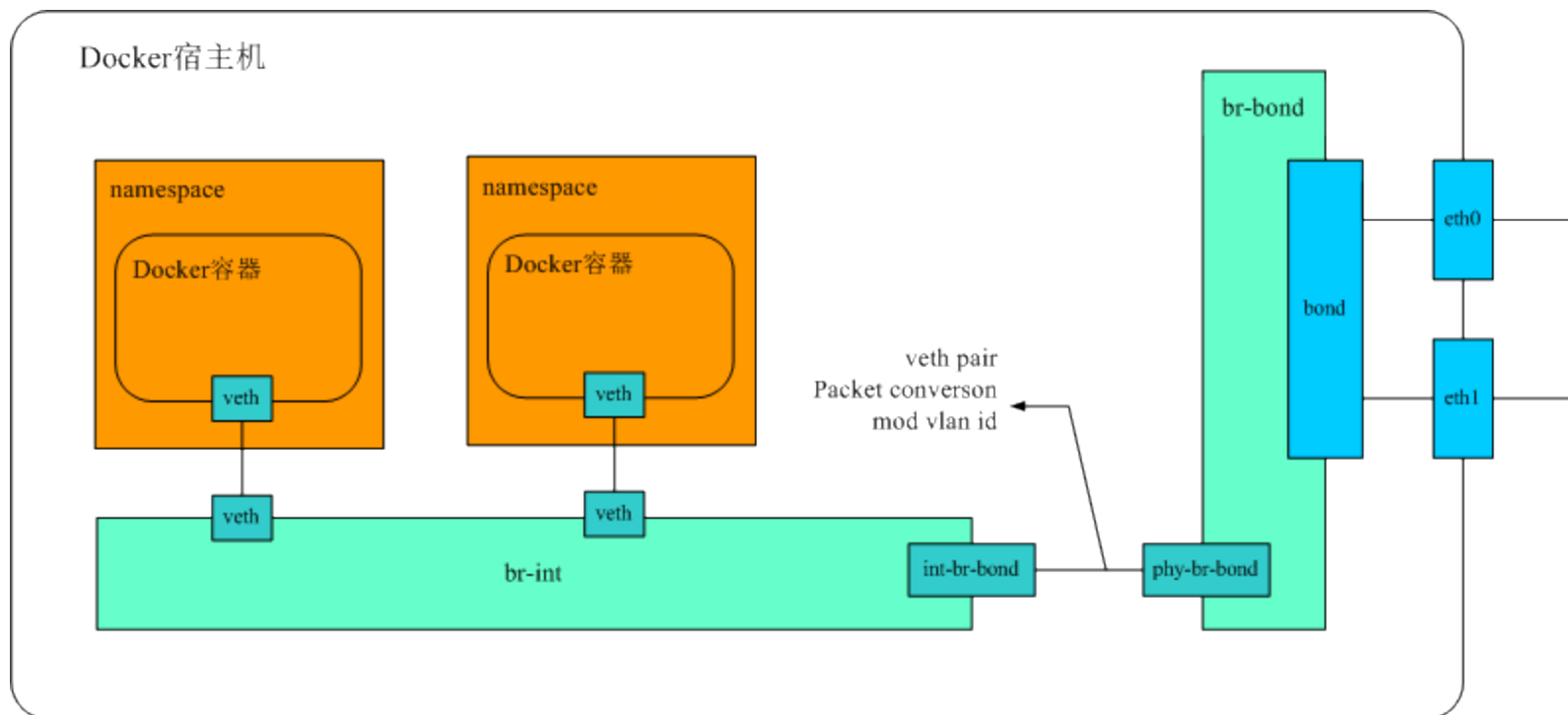
基于现有的OpenStack资源管理

通过Nova Docker Driver管理容器

容器的调度管理和VM一致



OpenStack管理容器的网络 等同于VM的VLAN网络模型



VM模式，应用额外部署

- 高度接近VM环境

目前支持发行版

- CentOS6.4: upstart
- CentOS7.1: systemd

容器权限

- `cap-drop mknod sys_boot sys_module sys_time`

参考VM镜像定制的环境

- 各种OPS的工具

直接从QCOW2的镜像中裁剪

- libguestfs导出rootfs
- Dockerfile 裁减不需要的内容
- 压缩镜像中间层

CentOS发行版

- upstart: 需要tty
- systemd: 需要cap_sys_admin, cgroup可写

容器化init

- runv
- s6
- supervisord

启动容器后配置port

- pipework、novadocker采用
- 容器启动后会有短暂时间没有网络
- Docker没有网络信息

libnetwork

weave

/proc 文件系统

- pid、net 命名空间隔离

sysinfo、sysconf 系统调用

- 没有命名空间隔离

Host监控

- docker stats
- cAdvisor
- zabbix-docker-monitoring

需要和现有方式整合

容器内监控

- 内核修改：/proc 文件系统
- 工具修改：top, vmstat
- lxcfs: 利用fuse提供修改过的/proc文件系统

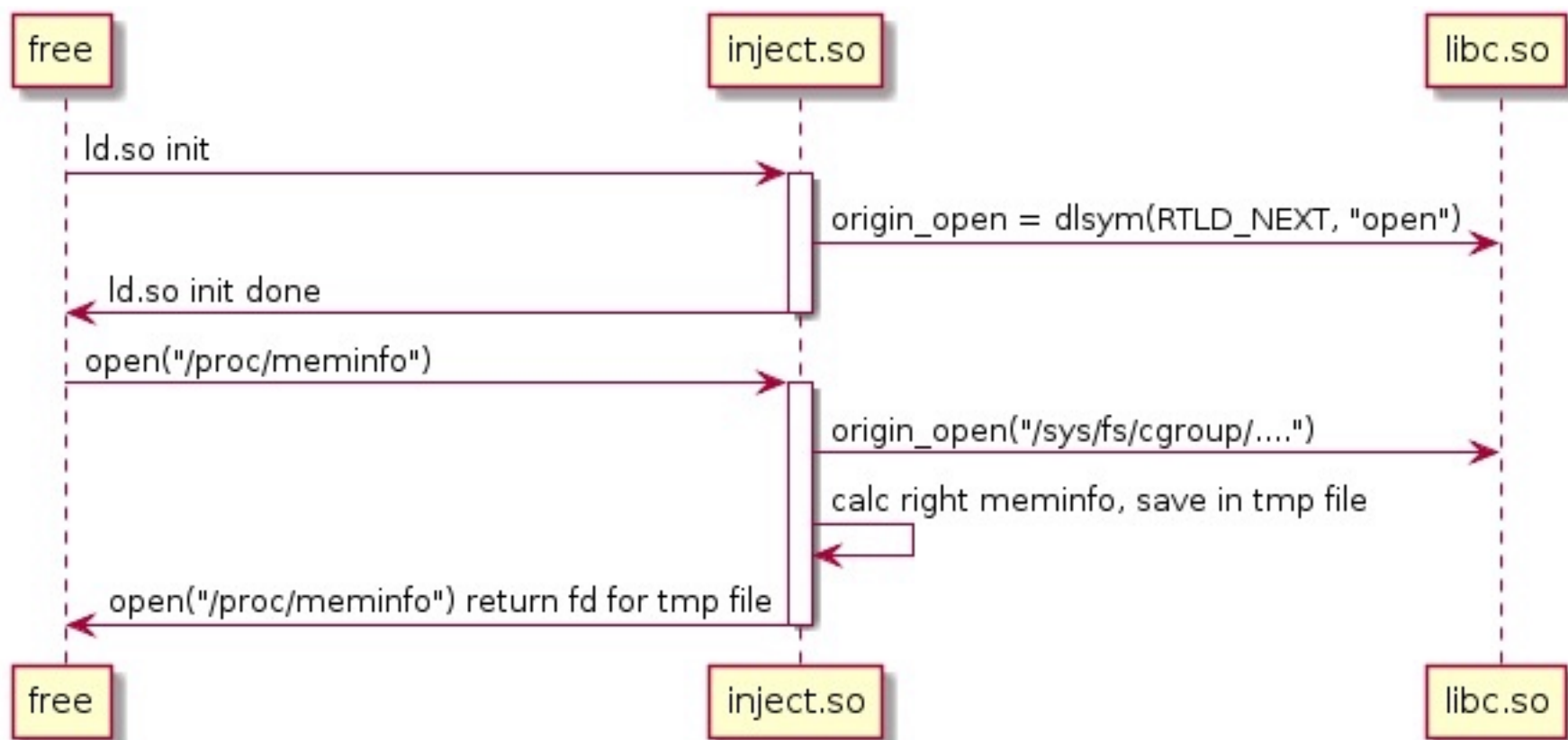
维护部署的代价

另一种容器内监控

- 通过LD_PRELOAD劫持对/proc文件的访问
- 参考lxcfs的实现重新计算/proc文件内容
- CPU load需要额外支持

```
[root@FATC1008 /]# free
              total        used         free       shared  buff/cache   available
Mem:           2148352      581012      691636      4680308      875704      691636
Swap:              0              0              0
[root@FATC1008 /]# unset LD_PRELOAD
[root@FATC1008 /]# free
              total        used         free       shared  buff/cache   available
Mem:          197881448      6580840     164832196      4680308     26468412     186118872
Swap:           8388604         32408      8356196
[root@FATC1008 /]#
```

另一种容器内监控



OpenStack in Docker

基于镜像的持续交付

Windows Container

容器动态调度

谢谢

