如何构建易用的Lustre系统管理软件

李希

如何让Lustre变得易用？

- 当前Lustre的使用难点
  - 学习曲线陡峭、概念众多、功能庞杂
  - Lustre文件系统只提供数据存储能力，不集成周边软件系统
  - 周边软件庞杂、分散，难以有机组合
- 用户的期望
  - 开箱即用：不用手动安装配置一堆软件
  - 含义明确：不用反复查阅Lustre用户手册
  - 界面友好：不用直面数十个各类命令和参数
  - 适用于集群：不用在众多的MDS/OSS上手动管理数十个MDT/OST
  - 技术支持完善：不用独自摸索、试错

Coral：易用的集成化Lustre开源发行版

COFS China Open File System

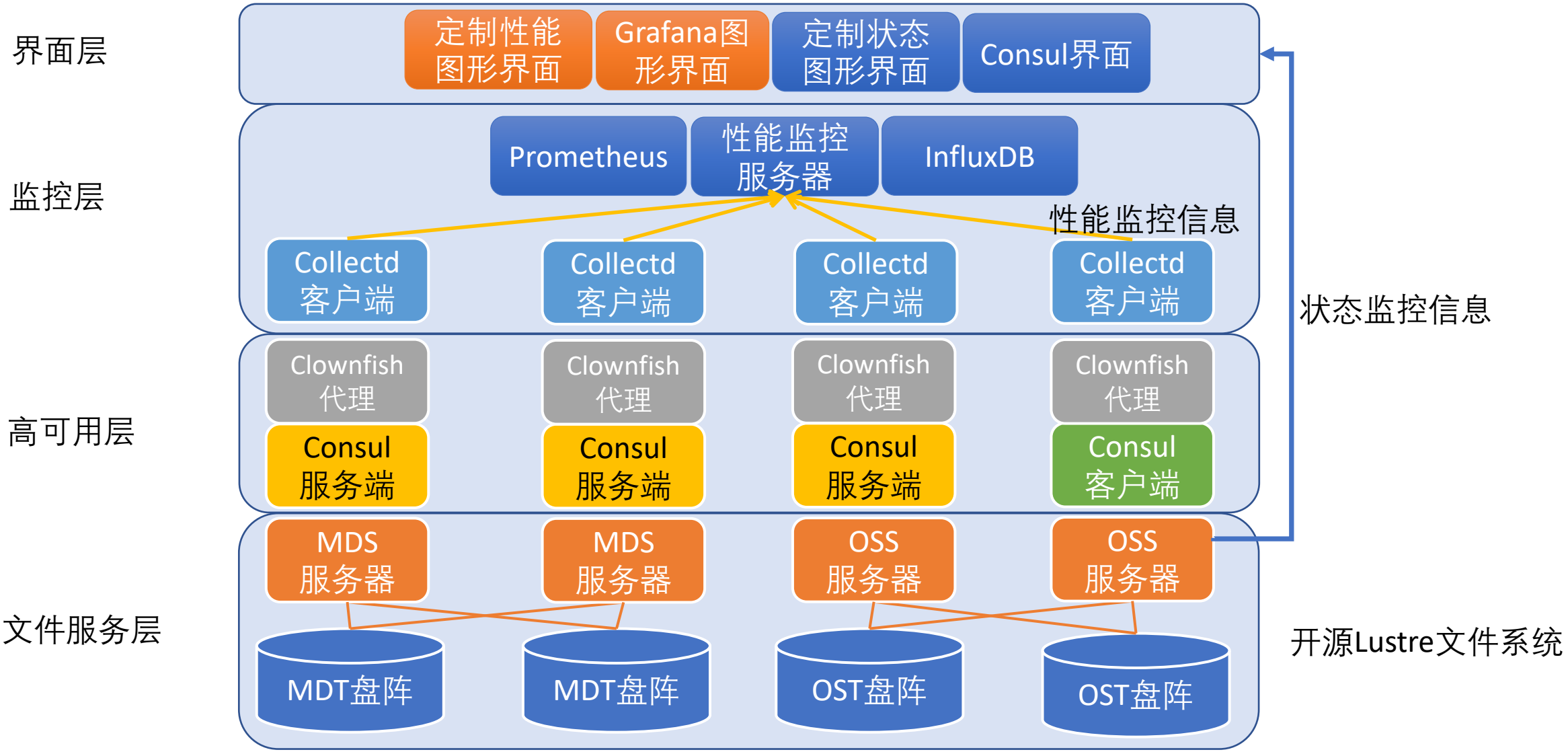
- 历经长期的技术积累与演化（~10年）
  - LustrePerfMon、Barreleye、Clownfish、Lime
- 集成Lustre文件系统开源发行版
  - 免去用户选取和编译Lustre的困扰
  - 可进行额外的测试和验证
  - 可进行额外的硬件适配和调优
- 集成各类Lustre周边软件功能
  - 高可用、系统管理、配置管理、状态监控、性能监控
- 组件化设计与实现
  - 现已集成部件coral-clownfish和coral-barreleye
  - 部件之间有机结合，避免功能和代码重叠
  - 各功能组件可独立安装部署，也可组合使用
  - 软件系统结构可扩展，不断拓展增加新组件

Coral:集成化的Lustrer软件发行版

Clownfish:基础管理

Barreleye:性能监控

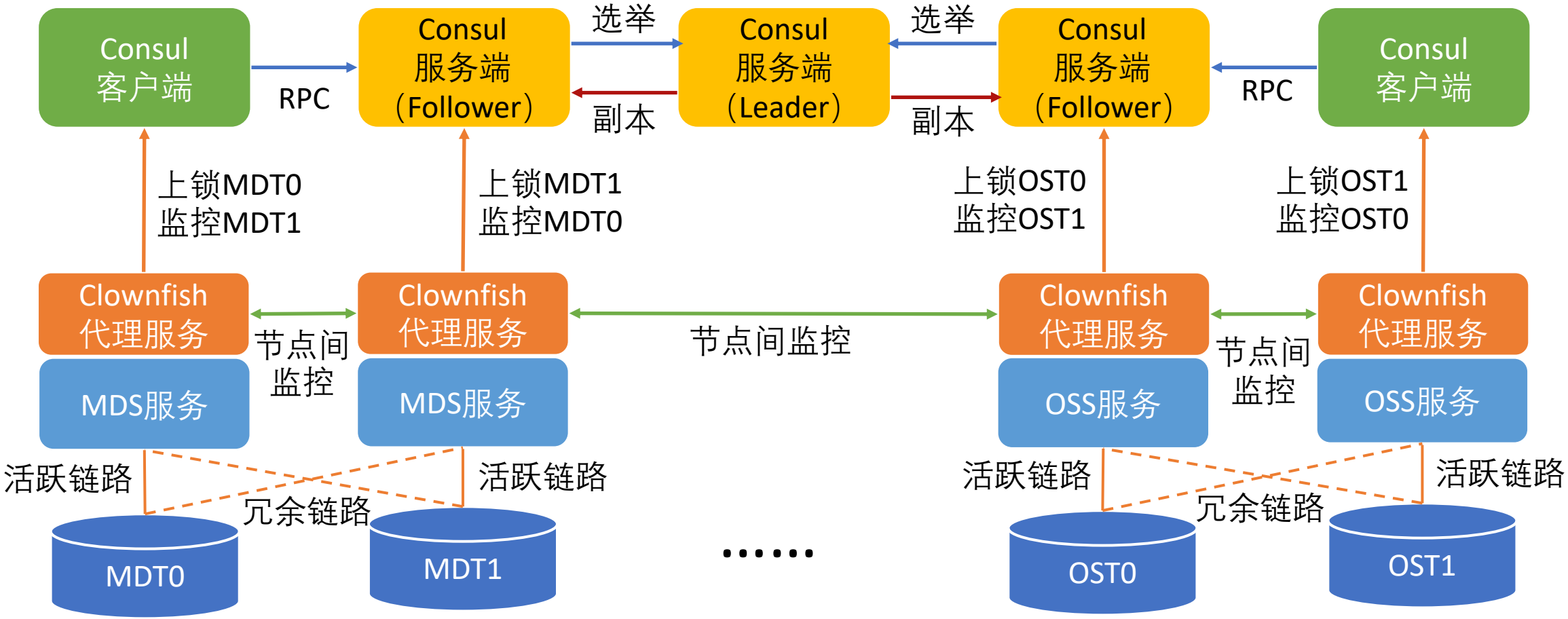
Coral基础软件栈的结构原理



现有Lustre高可用机制实现方式的不足

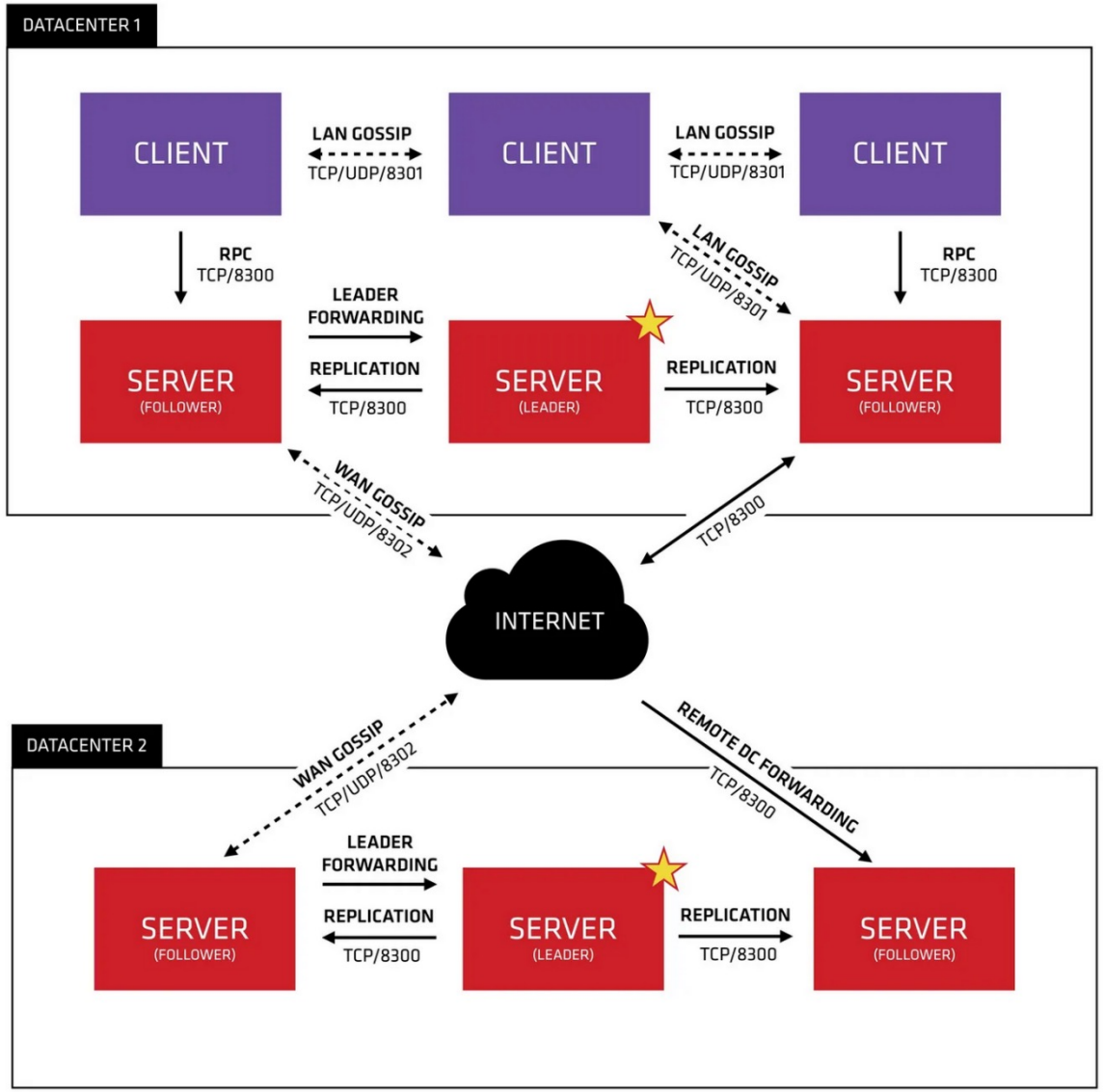
- 基于Corosync/Pacemaker的Lustre HA解决方案
  - 优点：技术成熟、使用广泛、适合小规模集群和简单系统
  - 不足：难以自动化部署和配置，不适合状态复杂度高的Lustre系统，可扩展性存在限制（集群节点的上限为16个）
- 高可用软件引发众多问题
  - Lustre文件系统软件本身的可靠性得到很大提升（2.12+）
  - 高可用软件自身问题：脑裂、失去同步、高可用节点对同时重启对方
  - 高可用软件触发和激化Lustre问题：Lustre尚未恢复就被高可用软件重启、数据损坏
  - 高可用引起的存储系统问题在软件问题中占比很大（~30%）

Coral-Clownfish的高可用机制的原理



Consul的原理

- Consul的结构特点
  - 分布式：适合支持包含众多服务节点的Lustre集群
  - 高可用：可支撑Lustre高可用机制的各方面需求
  - 可横向扩展：可支撑Lustre添加服务节点和存储目标的需求
- 可供Coral/Lustre集群利用的Consul特性
  - 服务发现：注册各类Coral服务，以供其他服务发现和利用（暂未利用）
  - 健康检查：自定义Lustre节点和服务的监控检查方法，监视集群的健康状况（暂未利用）
  - Key/Value存储：动态配置、功能标记、领袖选举（已利用）
  - 多数据中心：将Coral的配置和监控数据同步到多个数据中心（暂未利用）



Coral-Clownfish的实现特点

- 利用Consul/ZooKeeper/etcd等新型高可用软件替换Corosync/Pacemaker
- 利用Consul的Key/Value存储，对Lustre存储系统进行动态配置
- 利用Consul的图形界面实现配置的图形化管理
- 所有的软件包都包含在单一ISO中，如coral-2.0.1.el7.x86_64.iso
  - 集成最新长线支持版开源Lustre RPM
  - 集成Lustre相匹配的E2fsprogs RPM
  - 集成Coral的Clownfish和Barreleye的RPM
  - 集成依赖的所有软件包，可在最简安装的CentOS7服务器上无网络离线安装
- 一体化的管理命令：clownf

采用Coral-Clownfish简化Lustre管理流程

COFS China Open File System

- 安装Coral软件包

```
# mount -o loop coral-*.iso /mnt/iso/
# rpm -ivh /mnt/iso/Packages/coral-*
```

- 编辑Clownfish配置
  - /etc/coral/clownfish.conf单文件存储了关于Clownfish集群的所有配置信息，包含：Lustre文件系统名、各存储设备、集群内各服务节点、存储网络配置等
  - /etc/coral文件下有配置实例供参考

```
# cp /etc/coral/clownfish.conf.example /etc/coral/clownfish.conf
```

- 安装和配置Clownfish集群
  - 在集群所有节点上安装Coral软件包、同步Clownfish配置、安装和配置Consul软件、构建Clownfish高可用集群
  - 不包含Lustre软件的安装

```
# clownf cluster install --iso coral-*.iso
```

- 在Clownfish集群上安装Lustre软件
  - 在集群内所有Lustre节点上安装Lustre服务器内核、启动内核、安装Lustre软件包

```
# clownf cluster prepare
```

- 格式化Lustre文件系统
  - 按/etc/coral/clownfish.conf的配置格式化Lustre文件系统的MGS、MDT和OST

```
# clownf cluster format
```

- 挂载Lustre文件系统

```
# clownf cluster mount
```

集群操作命令：clownf cluster

- 命令格式：插件 对象 操作
- clownf cluster status
  - 查询集群的负载均衡和监控状况
- clownf cluster autostart_[disable|enable|status]
  - 打开/关闭/查看集群内所有节点和服务的自动启动
- clownf cluster [fs|clients|services|hosts] [--status]
  - 列出集群内所有的文件系统/客户端/服务/节点
- clownf cluster install [--iso $CORAL_ISO]
  - 在集群内所有节点上安装和配置coral-clownfish软件，包括安装和设置Consul及高可用机制
- clownf cluster prepare
  - 在集群内的所有节点上安装Lustre软件及相关依赖
- clownf cluster [format|mount|umount]
  - 格式化/挂载/卸载集群内的所有MDS、MDT、OST
- clownf cluster [watchers|waching]
  - 查看集群内的所有节点和服务的高可用监控者和被监控者状态

文件系统操作命令：clownf fs

- clownf fs ls [--status]
  - 列出所有的文件系统
- clownf fs autostart_[disable|enable|status] <fsname>
  - 打开/关闭/查看文件系统所有节点和服务的自动启动设置
- clownf fs [clients|services|hosts] [--status] <fsname>
  - 列出文件系统的所有客户端/服务/节点
- clownf fs [format|mount|umount] <fsname>
  - 格式化/挂载/卸载某个文件系统的所有MDS、MDT和OST
- clownf fs [watchers|waching] <fsname>
  - 查看文件系统的所有节点和服务的高可用监控者和被监控者状态

服务操作命令：clownf service

COFS China Open File System

- clownf service ls [--status]
  - 列出所有的MGS/MDT/OST服务
- clownf service status <service_name>
  - 查看指定Lustre MGS/MDT/OST服务的状态
- clownf service autostart_[disable|enable|status] <service_name>
  - 打开/关闭/查看某MGS/MDT/OST服务的自动启动设置
- clownf service host_[disable|enable|status] <service_name> <host>
  - 允许/不允许某MGS/MDT/OST服务挂载在某节点上
- clownf service hosts [--status] <service_name>
  - 列出某Lustre MGS/OST/MDT服务允许挂载的所有节点
- clownf service [format|mount|umount] <service_name>
  - 格式化/挂载/卸载某Lustre MGS/MDT/OST服务
- clownf service watcher <service_name>
  - 查看某Lustre MGS/MDT/OST服务所有高可用监控者的状态

节点操作命令：clownf host

COFS China Open File System

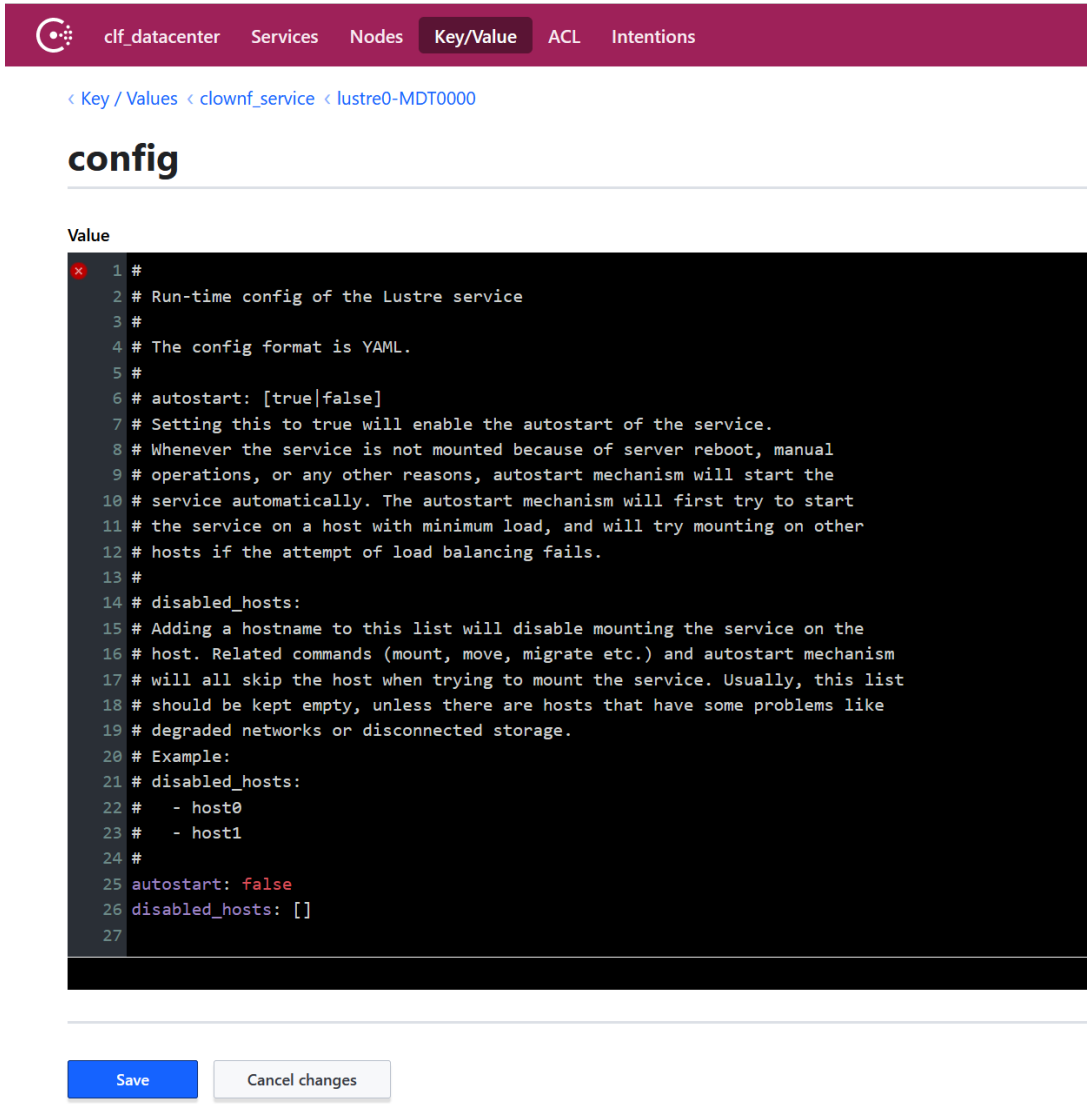
- clownf host ls [--status]
  - 列出所有的节点
- clownf host status <hostname>
  - 查看指定Lustre MDT/OST服务的状态
- clownf host autostart_[disable|enable|status]
  - 打开/关闭/查看节点的自动启动设置
- clownf host install <hostname>
  - 在指定节点上安装和配置coral-clownfish软件，包括安装和设置Consul及高可用机制
- clownf host prepare <hostname>
  - 在指定节点上安装Lustre软件及相关依赖
- clownf host services_[disable|enable]
  - 允许/不允许在某节点上挂载任何Lustre MGS/OST/MDT服务
- clownf host services_migrate
  - 将指定节点上的所有已挂载的Lustre MGS/OST/MDT服务迁移到其他节点
- clownf host services [--status]
  - 查看指点节点上所有可挂载的Lustre MGS/OST/MDT服务状态
- clownf host [shutdown|start]
  - 关闭/启动指点节点
- clownf host [watcher|watching]
  - 查看某节点所有高可用监控者/被监控者的状态

客户端操作命令：clownf client
Consul操作命令：clownf consul

- clownf client ls [--status]
  - 列出所有的Lustre客户端
- clownf client status <client_name>
  - 查看指定Lustre客户端的状态
- clownf client [mount|umount] <client_name>
  - 挂载/卸载指定的Lustre客户端
- clownf consul status
  - 查看Consul集群的所有成员节点的状态
- clownf consul leader
  - 查看Consul的leader节点
- clownf consul members
  - 查看Consul的所有成员节点
- clownf consul reset
  - 复位Consul的配置和数据
- clownf consul [start|restart]
  - 启动/重启动集群内的所有Consul服务

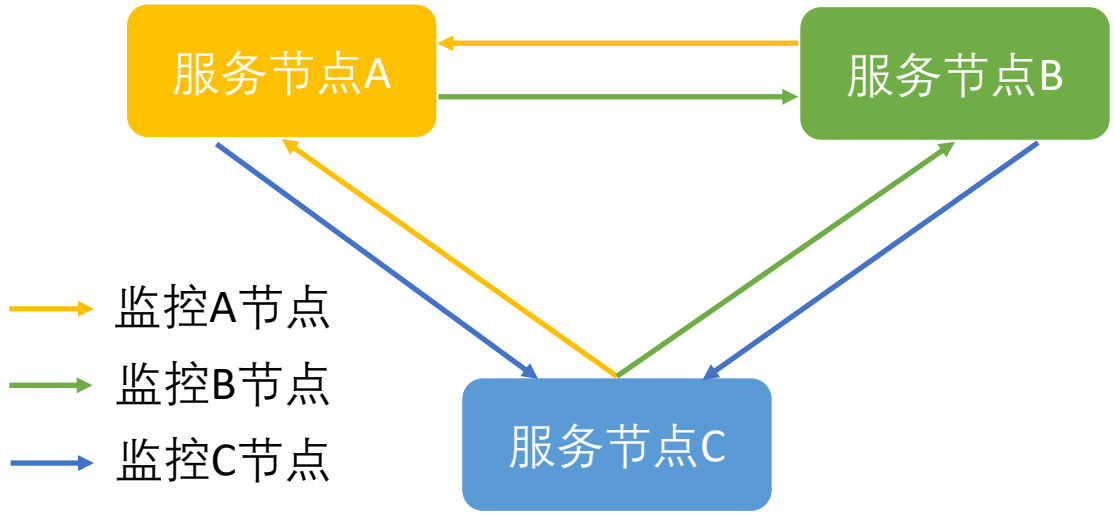
图形界面：Lustre服务配置管理

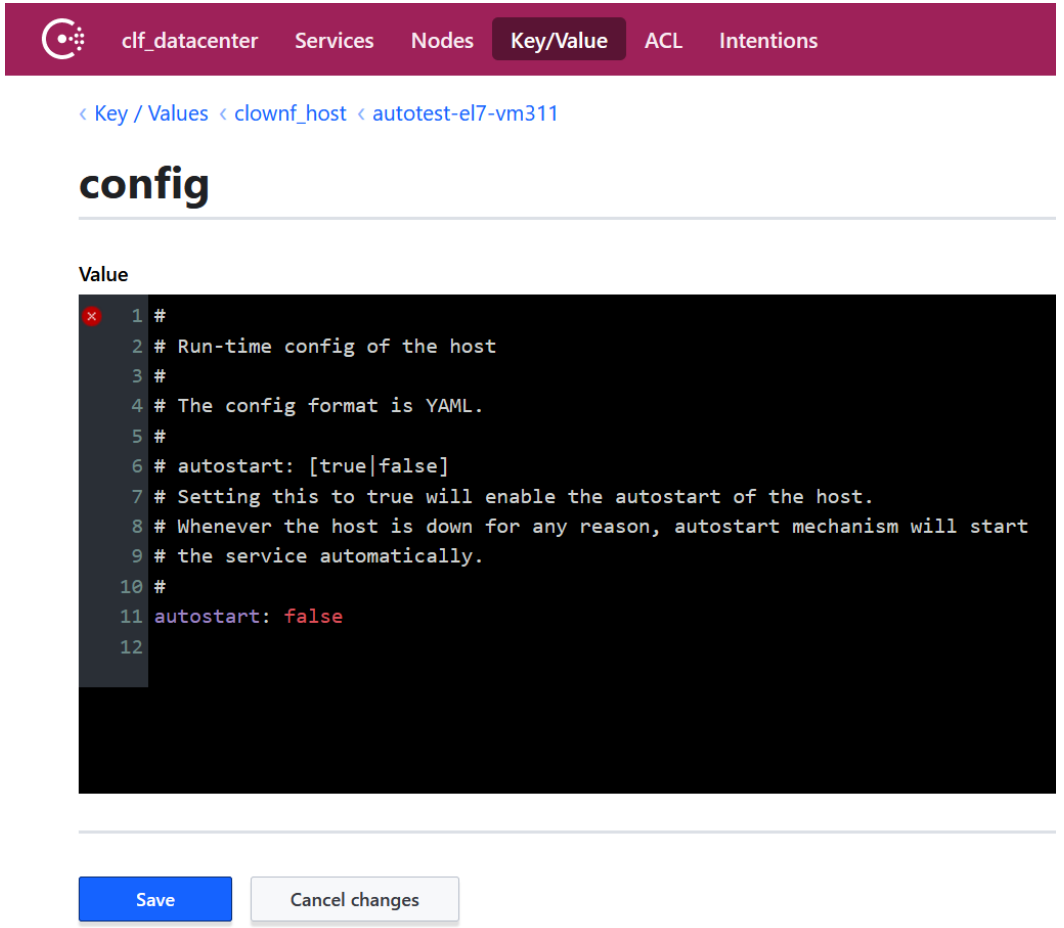
- 管理Lustre OST/MDT的高可用配置
  - 更改的配置会利用Consul机制在集群内立即同步和生效
  - Consul的多副本机制保证配置本身的高可用
- autostart：在发现OST/MDT服务未正常挂载时是否自动挂载
- disabled_hosts：避免将OST/MDT挂载在哪些节点上
  - 负载均衡、网络/存储连接不对称、绕开故障节点
- 后续可逐步添加对其他Lustre服务相关配置的支持
  - 服务首选的挂载节点
  - 是否允许服务自动迁移（负载均衡）



图形界面：服务节点配置管理

- 管理Lustre服务节点的高可用配置
- 可设置在发现服务节点宕机后是否自动启动服务器
- 支持采用IPMI接口自动启动服务节点
- 支持基于KVM的虚拟服务器高可用

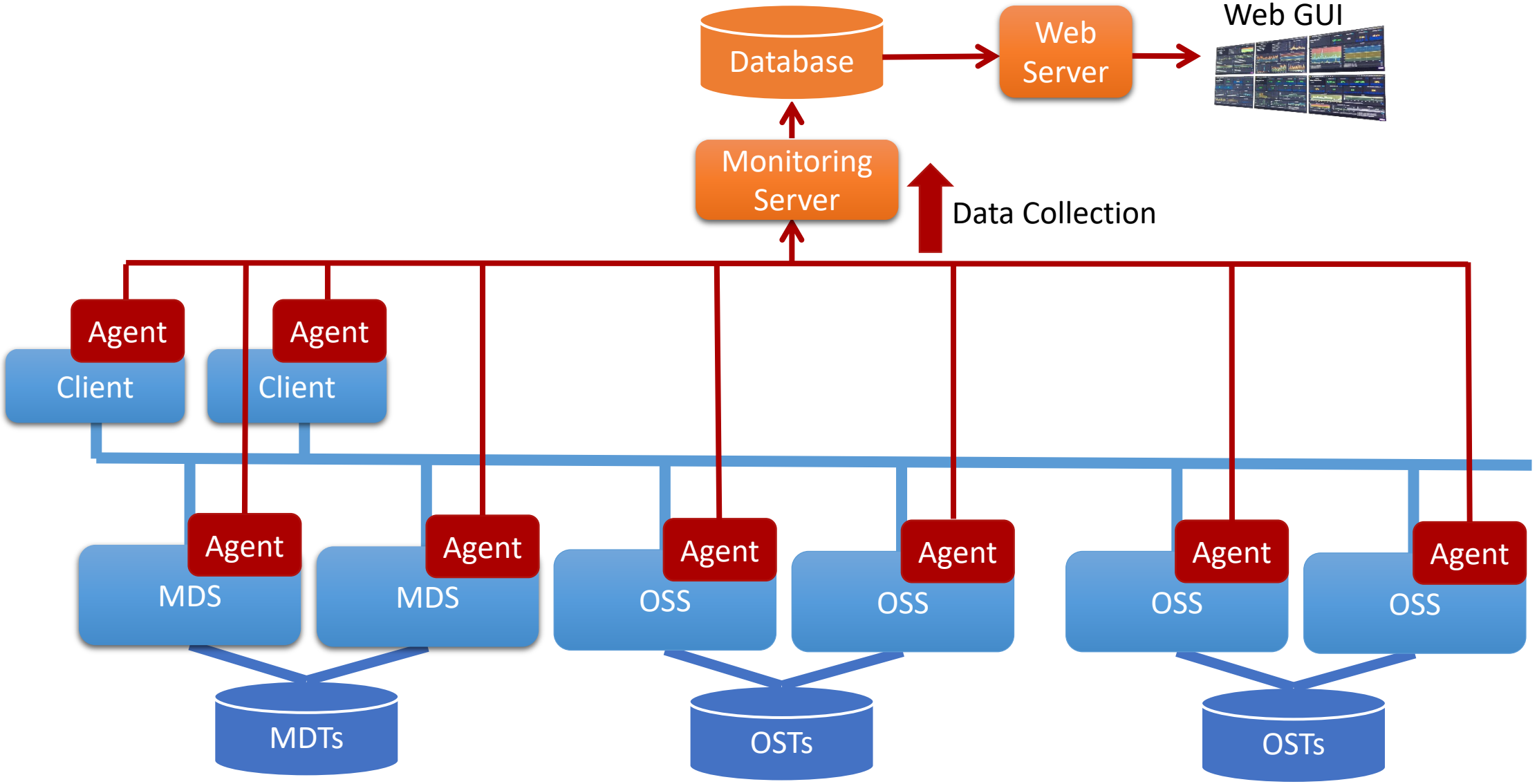




Coral-Barreleye： Lustre性能监控系统特性

- Collectd-5.12 + 深度定制的Lustre数据收集插件
- Lustre数据格式的XML描述文件
- Grafana-7.3.7 + 定制的Lustre仪表盘
- 时序数据库Influxdb-1.8.4
- 一体化管理命令行： barrele
- 支持Lustre版本2.7、2.10、2.12和2.14
- 支持CentOS7、CentOS8、Ubuntuu
- 支持X86_64和ARM架构

Coral-Barreleye性能监控系统的架构



Coral-Barreleye可监控的数据内容

- 服务器基本数据:
  - CPU、内存、根文件系统使用率、启动时间、服务器负载、温度、在线用户数目
- 存储盘阵基本数据:
  - 磁盘读写延迟、磁盘IOPS、磁盘吞吐率、磁盘请求I/O粒度
- Lustre文件系统综合数据:
  - Inode使用量、空间使用量、文件系统吞吐率、IOPS
- Lustre元数据存储服务器（MDS）上的监控数据:
  - 数据读写RPC大小、请求队列长度
- Lustre对象存储服务器（OSS）上的监控数据:
  - 元数据速率、请求队列长度
- Lustre客户端上的监控数据:
  - 数据/元数据访问速率、访问延迟

Coral未来展望：构建Lustre文件系统综合软件栈

- 可靠的基础系统管理工具
  - 系统管理工具集、高可用功能、状态监控、性能监控、限额管理等
- 全面的数据访问接口
  - S3、NFS、Samba、K8S、OpenStack Cinder
- 易用的功能特性
  - 如安全保护、多租户、审计等
- 强大的数据管理能力
  - 数据迁移、元数据迁移、用量均衡、冷热池管理、分级存储管理等
- 专业的生产工具
  - 回归测试、功能验证、性能瓶颈分析等

Coral项目的近期开发方向

- Barreleye
  - 采用数据库VictoriaMetrics替换InfluxDB
  - 对Lustre-2.15的支持
  - 单独升级Grafana Dashboard的命令：`barrele dashboard upgrade`
- Clownfish
  - 设计和实现添加新OST/MDT的顺畅流程和命令
  - 增加对OST挂载点偏好的支持：`clownf service bias_[set|clear|status]`
  - 实现缺失的管理命令，例如：`clownf fs prepare`
  - 简化/etc/coral/clownfish.conf配置文件
- 添加新的功能组件
  - Lustre配置管理
  - 状态监控

开源社区建设

- Coral源代码
  - https://github.com/LiXi-storage/barreleye
  - https://gitee.com/filesystem/barreleye
- 社区网站建设
  - 提供编译和测试完成的各版本Coral ISO
    - 各CPU体系结构平台
    - 各Lustre发行版本
    - 各Linux操作系统发行版本
    - 各Coral版本
  - 提供更加全面的文档和示例
- 更丰富的社区交流活动

谢谢！