



云平台数据接入模块的设计与实现

-王 斌

目录

- 数据表设计
- 数据接入模块结构设计
- MVC, Singleton设计模式
- I/O复用 epoll模型(select, poll)
- 遇到的问题
- Git 的简单使用

数据表设计

id	sceneld	name	...	flag
1	10001	greenH ose	...	1
...				

场景表

id	deviceId	name	sceneld	flag
1	2000xx1	温度 (0x1)	10001	1
2	3000xx1	湿度 (0x2)	10001	1
3	2000xx2	温度 (0x1)	10002	1
4	3000xx2	湿度 (0x2)	10002	0
...				

设备表

id	deviceId	ctime	info	flag
1	2000xx1	3543455 346	varchar	1
2	...			

id	deviceId	ctime	info	flag
1	2000xx1	3543455 346	varchar	1
2	...			

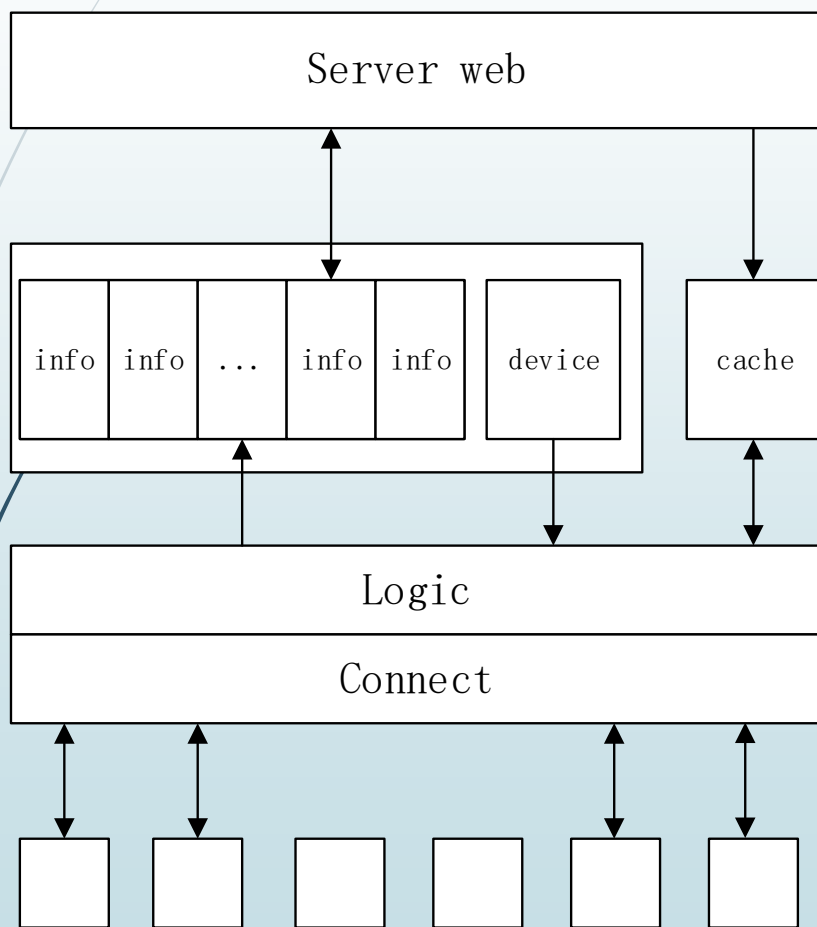
id	deviceId	ctime	info	flag
1	2000xx1	3543455 346	varchar	1
2	...			

id	deviceId	ctime	info	flag
1	3000xx2	3543455 346	varchar	0
2	...			

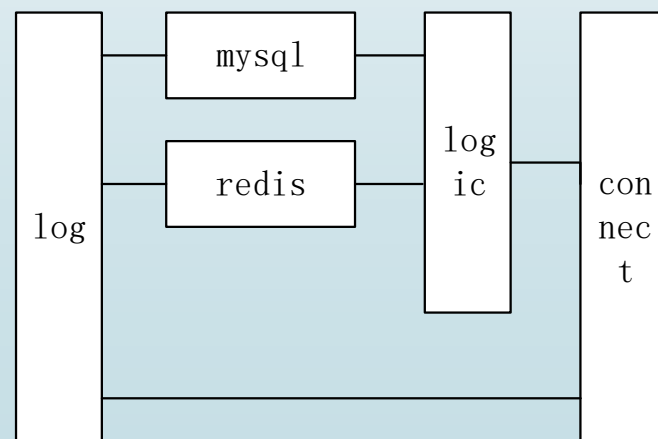
...

信息表

数据接入模块结构设计

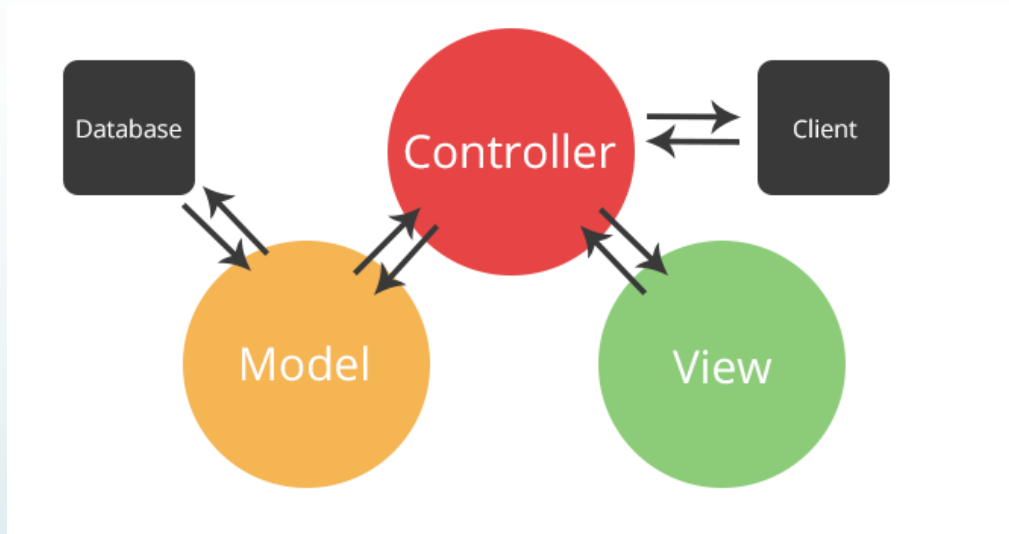


数据流图

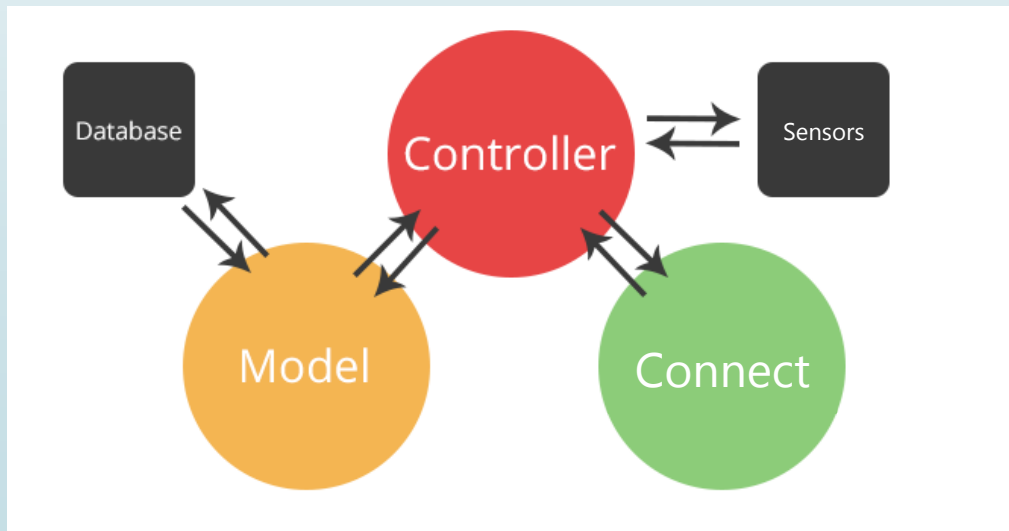


模块依赖关系图

MVC



MVC



MCC

单例

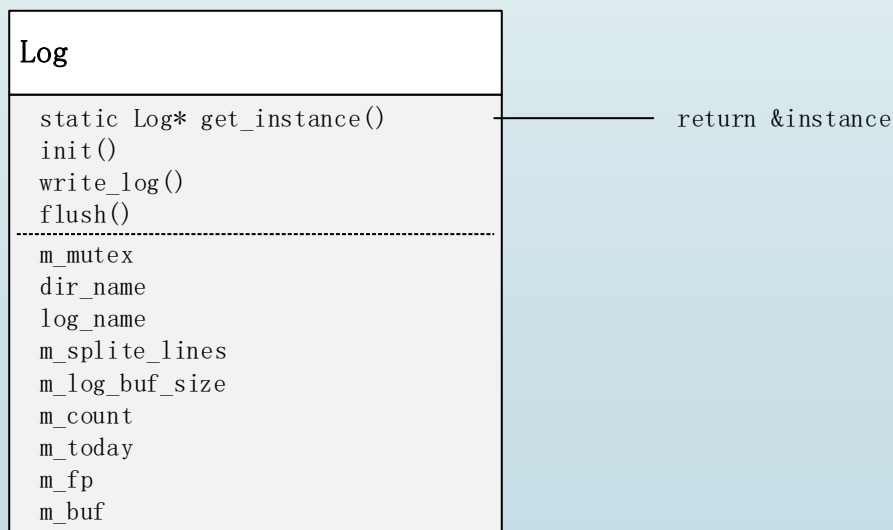
面对的问题：

- 每次写日志的时候都要初始化一个日志类对象；

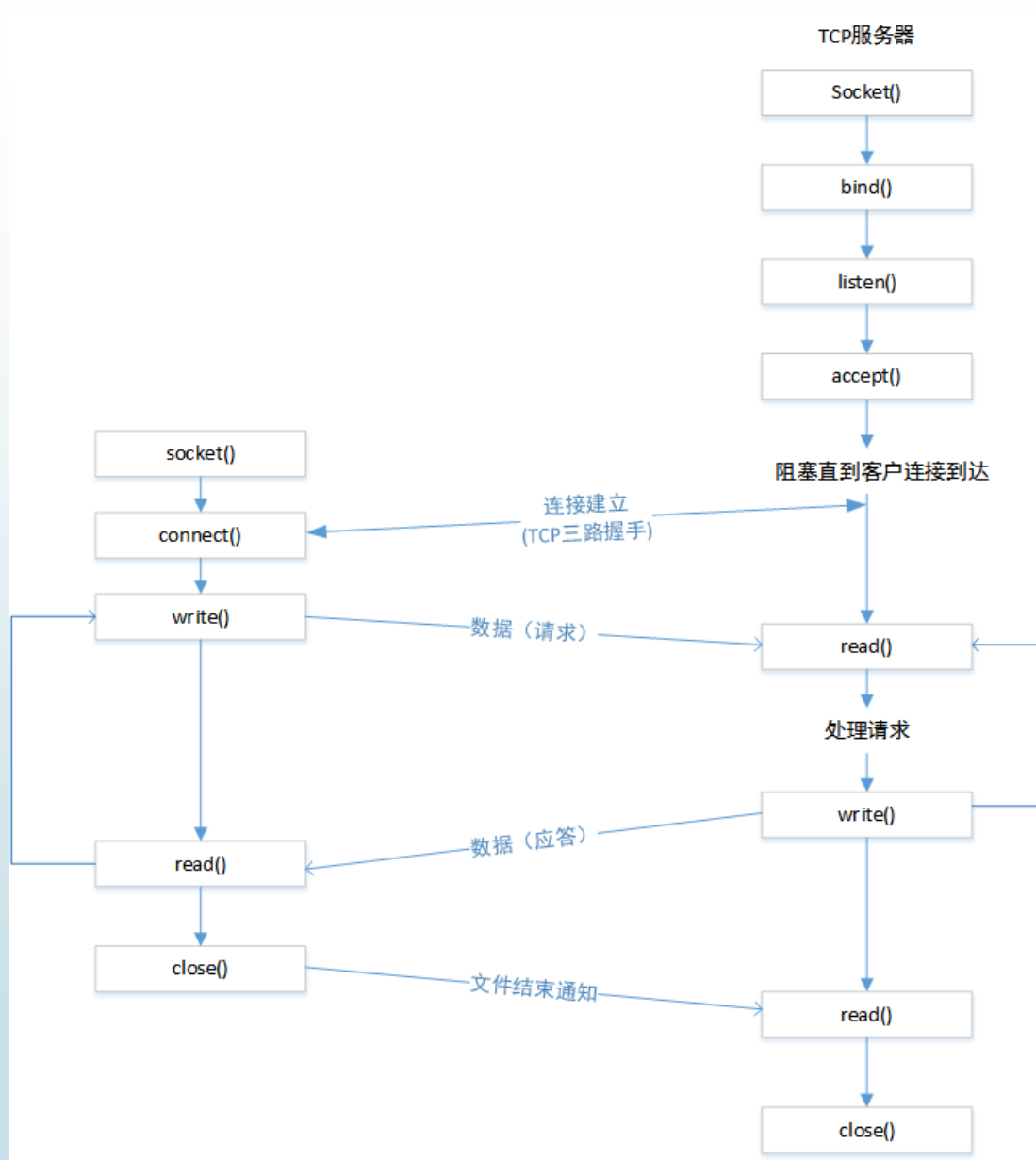
使用单例的目的：

- 服务端的日志信息存放在一个文件中，该文件由一个单例对象统一读取，并保证线程安全；
- 保证日志类仅有一个实例，并提供一个日志类的全局访问点；

实现：



日志模块UML类图



epoll

Epoll是什么

是类似于select、poll的I/O多路复用机制，一个进程可以监视多个描述符，一旦某个描述符就绪（读就绪或者写就绪），能够通知程序进行相应的读写操作。

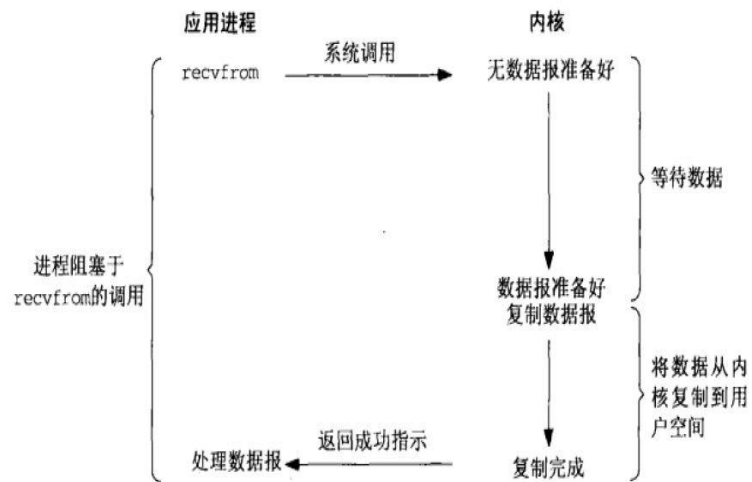


图6-1 阻塞式I/O模型

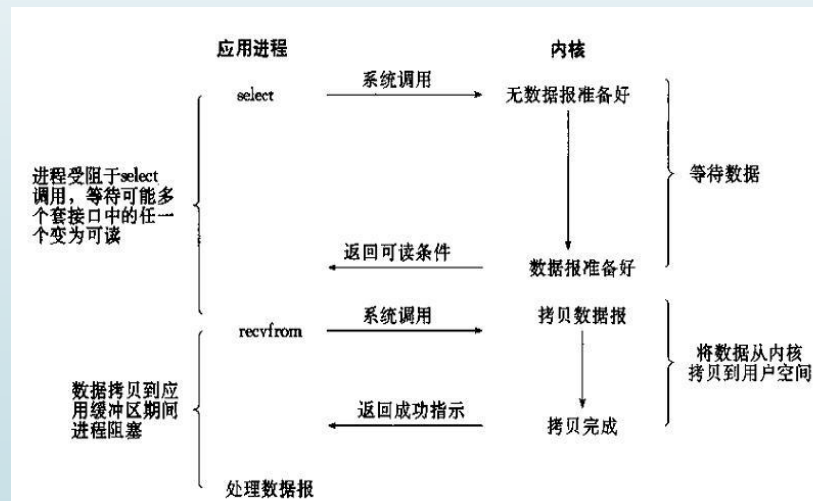


图 6.3 I/O 复用模型

epoll

Epoll是什么

epoll是在2.6内核中提出的，是之前的select和poll的增强版本。相对于select和poll来说，epoll更加灵活，没有描述符限制。epoll使用一个文件描述符管理多个描述符，将用户关系的文件描述符的事件存放到内核的一个事件表中，这样在用户空间和内核空间的copy只需一次。

Epoll操作接口

```
//创建一个epoll的句柄，size用来告诉内核这个监听的数目一共有多大
int epoll_create(int size);

//函数是对指定描述符fd执行op操作
//op操作，用三个宏来表示：添加EPOLL_CTL_ADD，删除EPOLL_CTL_DEL，修改EPOLL_CTL_MOD。
//分别添加、删除和修改对fd的监听事件。
int epoll_ctl(int epfd, int op, int fd, struct epoll_event *event);

//等待epfd上的io事件，最多返回maxevents个事件
int epoll_wait(int epfd, struct epoll_event * events, int maxevents, int timeout);
```



Epoll 执行流程



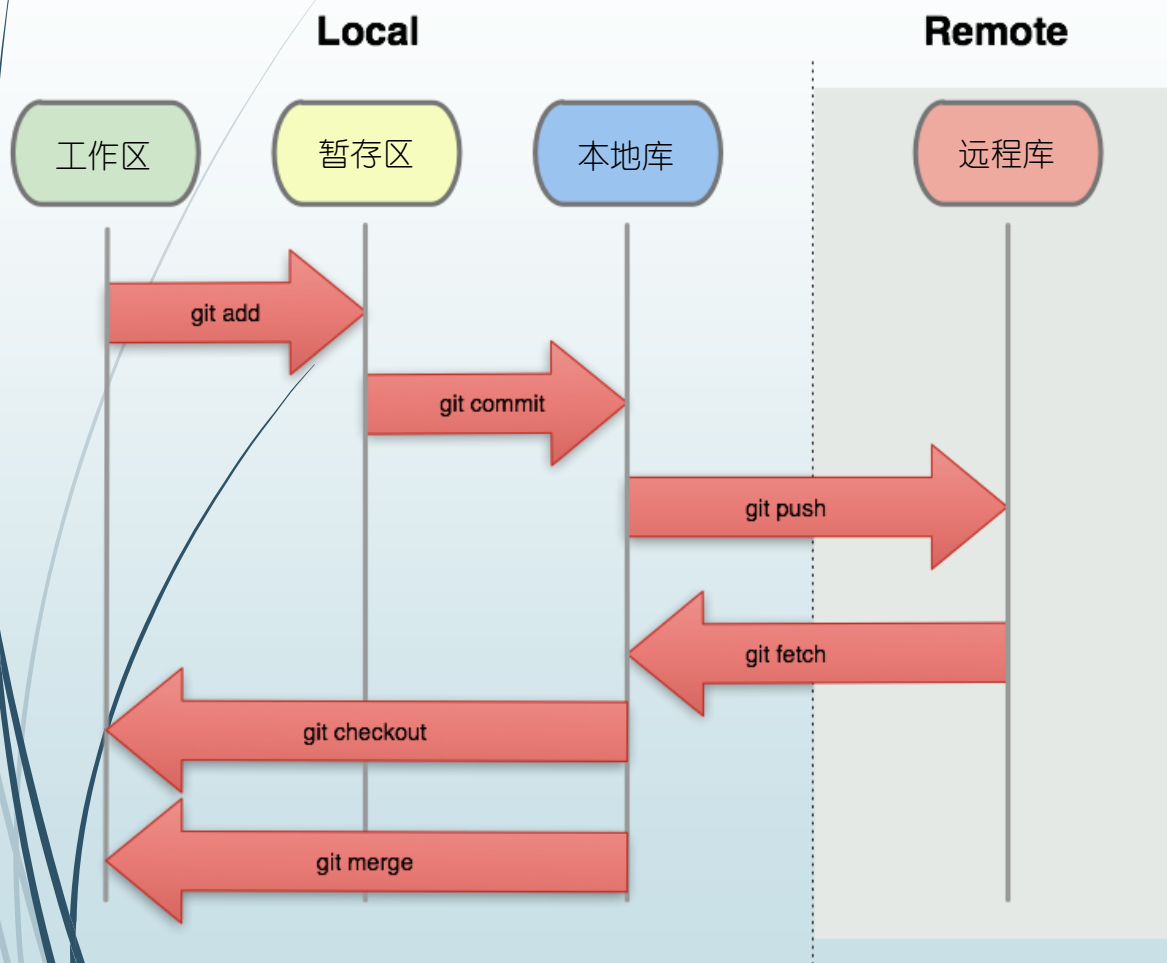
问题

遇到的问题：

- 日志文件不能同步刷新；
- 不同情况下写日志参数不一致；
- 客户端正与服务器通信时候**服务端**崩溃，服务端重启后客户端一段时间内连不上服务器；
- 客户端正客户端正与服务器通信时候**客户端**崩溃，服务端不知道客户端崩溃，导致连接资源一直占用；
- 服务器不够健壮，如epoll等待返回失败，等待超时可能造成服务程序直接退出；

遗留问题：

- 客户端超时问题解决的不严谨；
- 如果客户端与服务端是长连接通信，可能存在的粘包问题；
- 数据采集频率

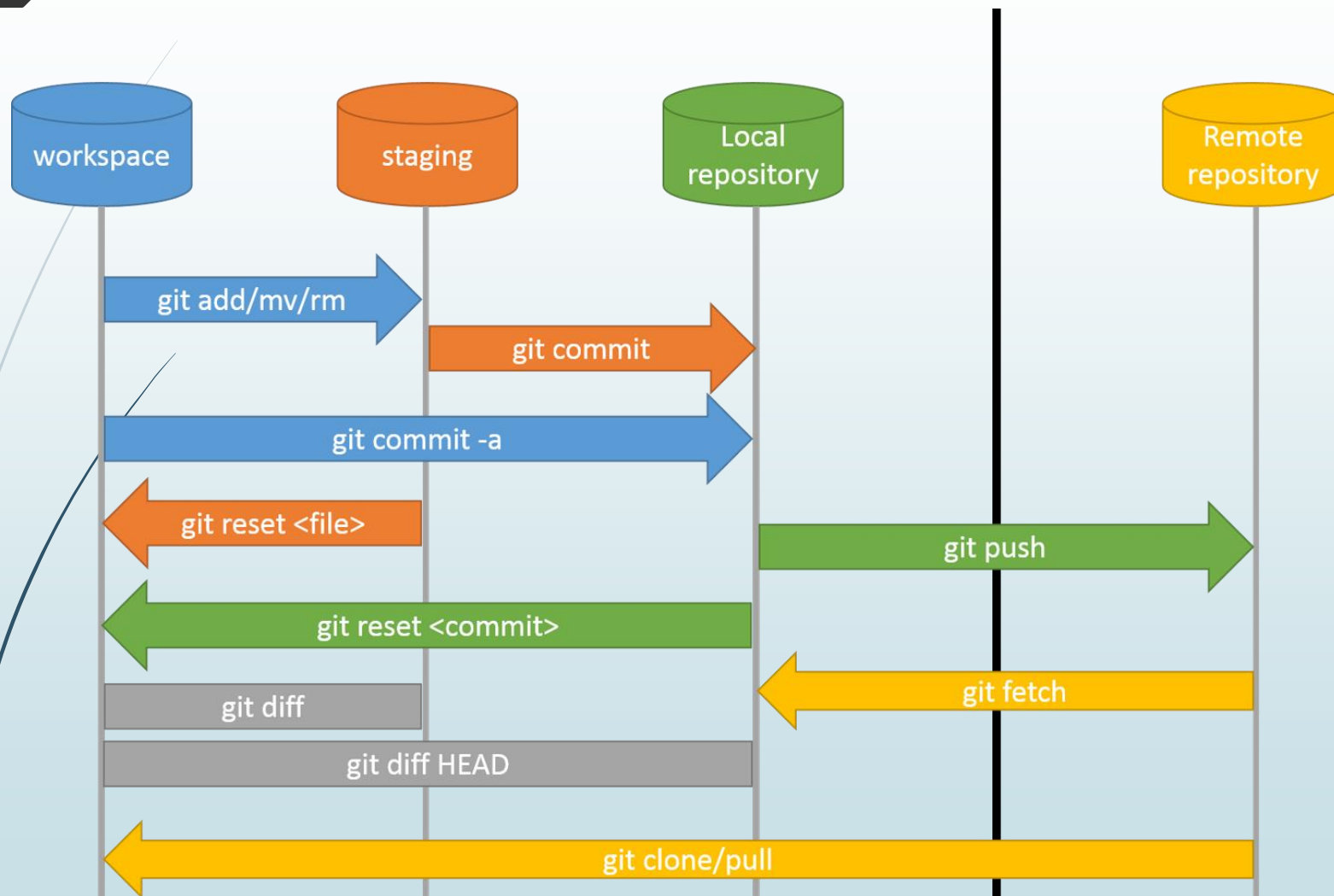


Git原理图

命令	操作内容
clone	从远程库下拉
add	添加到暂存区
commit	提交到本地库
push	推到远程库
status	查看当前状态
diff	比对
pull	同步
checkout	切换分支/清空暂存
branch	添加分支
fetch	从远程库拉到本地
merge	分支代码合并

常用命令表

git



参考资料

- Unix 网络编程-卷1
- 设计模式-可复用面向对象软件的基础
- Linux I/O模式及select, poll, epoll详解
(<https://segmentfault.com/a/1190000003063859>,
<http://blog.csdn.net/tianmohust/article/details/6677985>)
- Linux套接字编程中的5个隐患
(<http://blog.csdn.net/w746805370/article/details/72643341>)
- 廖雪峰的官方网站-git教程
(<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>)



THANKS!