

# 《分布式流媒体》实训项目

C/C++教学体系

# TNV DAY01

直播课

# 目录

概要设计

报文规约

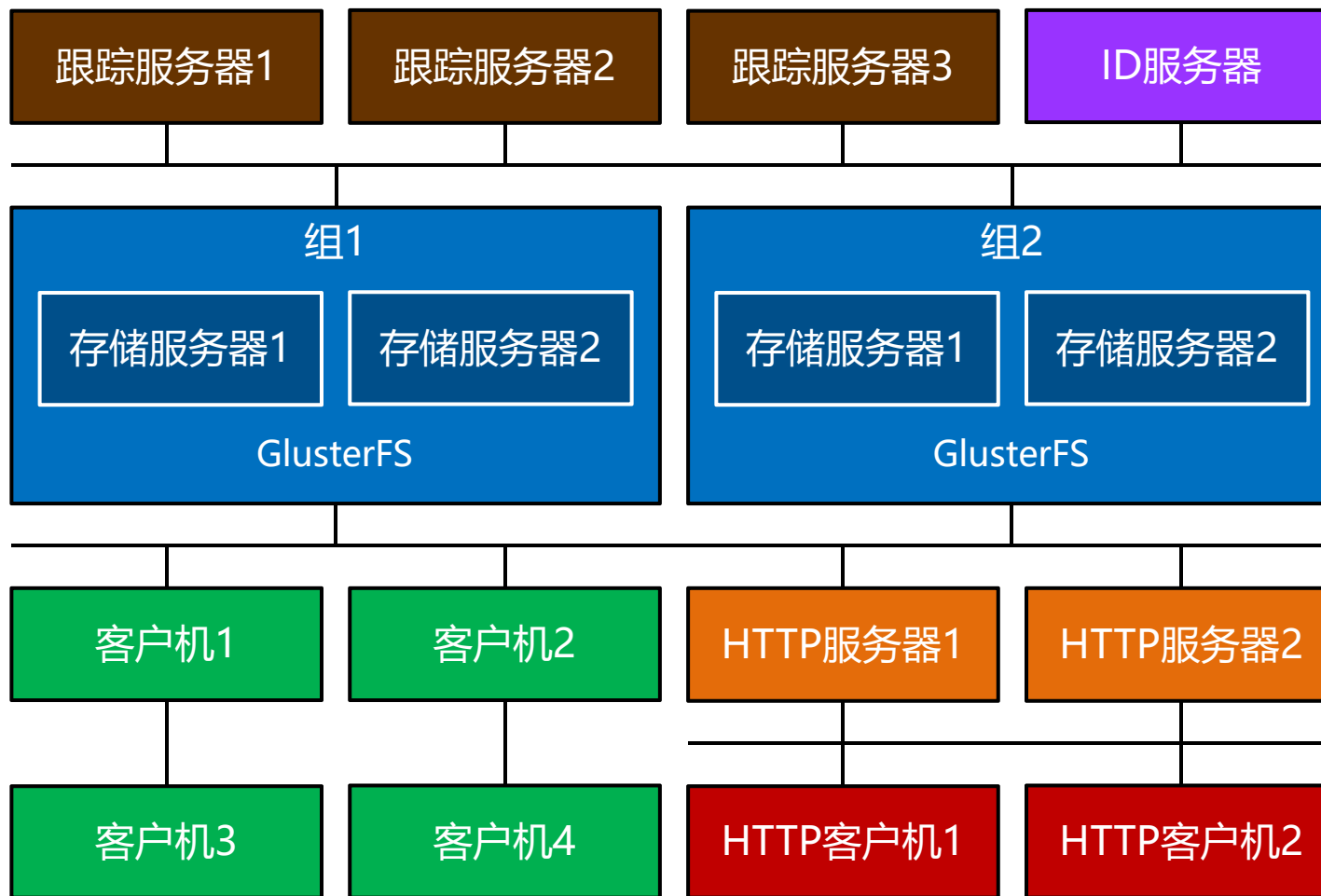
数据库设计

模块设计

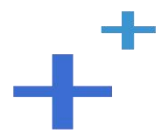
# 概要设计



# 总体架构



知识讲解



# 跟踪服务器

- 系统中包括一到多台跟踪服务器，每台跟踪服务器均维护完全相同的数据结构，以实时跟踪系统中每台存储服务器的状态变化
- 每台存储服务器启动伊始即将自己的详细信息加入到每台跟踪服务器所维护的数据结构中，并在此后以一定的周期间隔向每台跟踪服务器发出心跳信息，将自身的状态标记为活动
- 同时跟踪服务器也以一定的周期间隔检查其所维护的数据结构，将长期收不到心跳信息的存储服务器标记为离线

跟踪服务器1

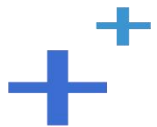
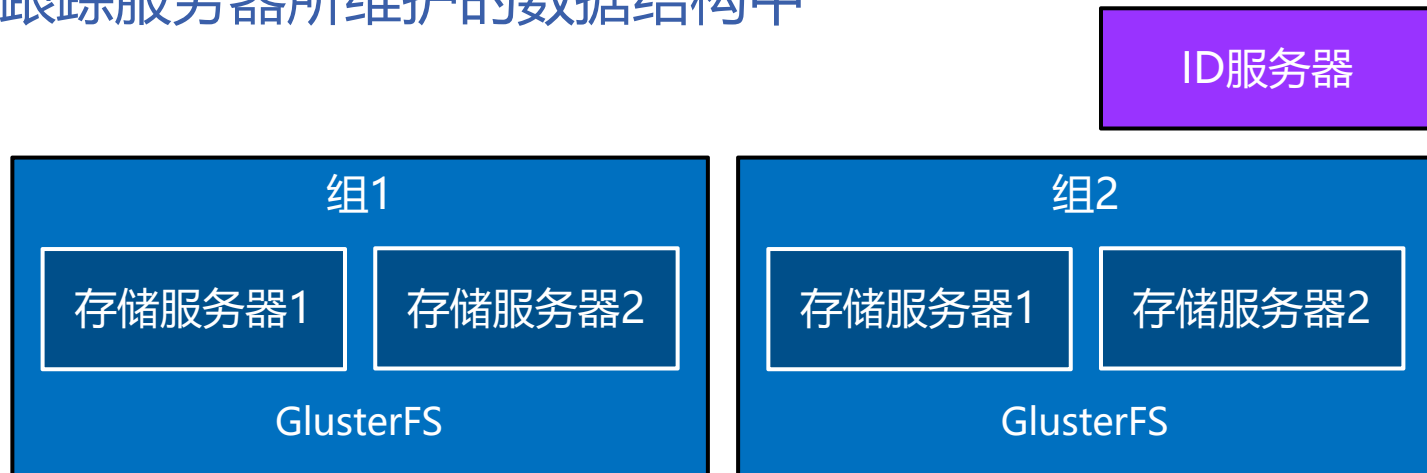
跟踪服务器2

跟踪服务器3



# 存储服务器和ID服务器

- 系统中包括一到多台存储服务器，它们被划分为若干组
- 每个存储服务器组通过GlusterFS保证其所有组员文件系统的同步性
- 每台存储服务器上的文件保存在特定的目录结构中，其存储路径根据取自ID服务器的唯一标识计算而得
- 每台存储服务器通过启动时的加入和周期性的心跳，保证其运行状态得以实时反映于每台跟踪服务器所维护的数据结构中



# 客户机

- 当某台客户机需要上传文件时
  - 会首先向任意一台跟踪服务器索要可用存储服务器的地址列表
  - 跟踪服务器根据客户机提交的用户ID，通过查询数据库确定其所隶属的组，并从其所维护的数据结构中挑选出若干台该组当前处于活动状态的存储服务器，以地址列表的形式返回给客户机
  - 而后客户机便可从此地址列表中选择一台存储服务器，发起上传文件请求
  - 后者一方面根据取自ID服务器的唯一标识确定存储路径并保存客户机上传的文件内容数据，另一方面将客户机提交的文件ID和该文件存储路径的对应关系保存在数据库中，以备日后下载和删除
  - GlusterFS将在稍后的时间里，把此文件同步到组内的其它存储服务器中





# 客户机

- 当某台客户机需要下载或删除文件时
  - 同样会先向跟踪服务器索要可用存储服务器地址列表，并向其中的一台存储服务器发起下载或删除请求
  - 存储服务器根据客户机提交的文件ID通过查询数据库找到与其对应的文件存储路径，读取其内容回传客户机或直接将其删除，对于后者还需要同时更新数据库，以反映存储中所发生的变化
  - GlusterFS也同样会把这些变化同步到组内的其它存储服务器中

客户机1

客户机2

客户机3

客户机4



# HTTP服务器和客户机

- HTTP服务器将客户机的下载功能包装在针对HTTP请求GET方法的处理中，直接将存储服务器下载得到的文件内容数据以HTTP响应的形式回传给HTTP客户机
- 后者可以是简单的wget命令，得到完整的离线媒体文件，也可以是任何支持HTTP协议的流媒体播放器，以实现在线实时播放

HTTP服务器1

HTTP服务器2

HTTP客户机1

HTTP客户机2



# 报文规约



# 存储服务器向跟踪服务器请求加入

- 请求报文

包头			包体					
包体长度	命令(10)	状态	版本	组名	主机名	端口号	启动时间	加入时间
8	1	1	包体长度					

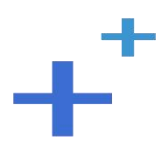
- 成功响应

包头		
包体长度	命令(100)	状态
8	1	1

- 失败响应

包头			包体	
包体长度	命令(100)	状态	错误号	错误描述
8	1	1	2	<=1024

知识讲解



# 存储服务器向跟踪服务器发送心跳

- 请求报文

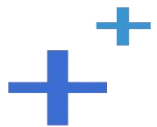
包头			包体	
包体长度	命令(11)	状态	组名	主机名
8	1	1	包体长度	

- 成功响应

包头		
包体长度	命令(100)	状态
8	1	1

- 失败响应

包头			包体	
包体长度	命令(100)	状态	错误号	错误描述
8	1	1	2	<=1024



# 客户机从跟踪服务器获取存储服务器地址列表

- 请求报文

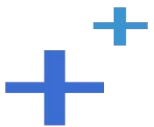
包头			包体		
包体长度	命令(12)	状态	应用ID	用户ID	文件ID
8	1	1	16	256	128

- 成功响应

包头			包体	
包体长度	命令(100)	状态	组名	存储服务器地址列表
8	1	1	包体长度	

- 失败响应

包头			包体	
包体长度	命令(100)	状态	错误号	错误描述
8	1	1	2	<=1024



# 客户机从跟踪服务器获取组列表

- 请求报文

包头		
包体长度	命令(13)	状态
8	1	1

- 成功响应

包头			包体
包体长度	命令(100)	状态	组列表
8	1	1	包体长度

- 失败响应

包头			包体	
包体长度	命令(100)	状态	错误号	错误描述
8	1	1	2	<=1024



# 存储服务器从ID服务器获取ID

- 请求报文

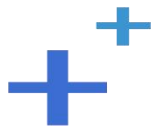
包头			包体
包体长度	命令(40)	状态	ID键
8	1	1	64+1

- 成功响应

包头			包体
包体长度	命令(101)	状态	ID值
8	1	1	8

- 失败响应

包头			包体	
包体长度	命令(101)	状态	错误号	错误描述
8	1	1	2	<=1024





# 客户机向存储服务器上传文件

- 请求报文

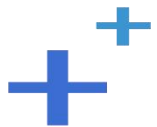
包头			包体				
包体长度	命令(70)	状态	应用ID	用户ID	文件ID	文件大小	文件内容
8	1	1	16	256	128	8	文件大小

- 成功响应

包头		
包体长度	命令(102)	状态
8	1	1

- 失败响应

包头			包体	
包体长度	命令(102)	状态	错误号	错误描述
8	1	1	2	<=1024



# 客户机向存储服务器询问文件大小

- 请求报文

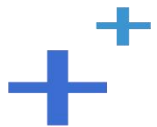
包头			包体		
包体长度	命令(71)	状态	应用ID	用户ID	文件ID
8	1	1	16	256	128

- 成功响应

包头			包体
包体长度	命令(102)	状态	文件大小
8	1	1	8

- 失败响应

包头			包体	
包体长度	命令(102)	状态	错误号	错误描述
8	1	1	2	<=1024



# 客户机从存储服务器下载文件

- 请求报文

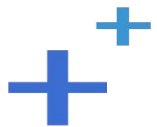
包头			包体				
包体长度	命令(72)	状态	应用ID	用户ID	文件ID	偏移	大小
8	1	1	16	256	128	8	8

- 成功响应

包头			包体
包体长度	命令(102)	状态	文件内容
8	1	1	大小

- 失败响应

包头			包体	
包体长度	命令(102)	状态	错误号	错误描述
8	1	1	2	<=1024



# 客户机要求删除存储服务器上的文件

- 请求报文

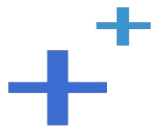
包头			包体		
包体长度	命令(73)	状态	应用ID	用户ID	文件ID
8	1	1	16	256	128

- 成功响应

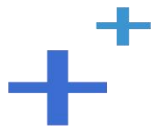
包头		
包体长度	命令(102)	状态
8	1	1

- 失败响应

包头			包体	
包体长度	命令(102)	状态	错误号	错误描述
8	1	1	2	<=1024



# 数据库设计



# 三库六表

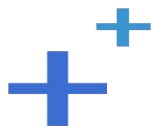
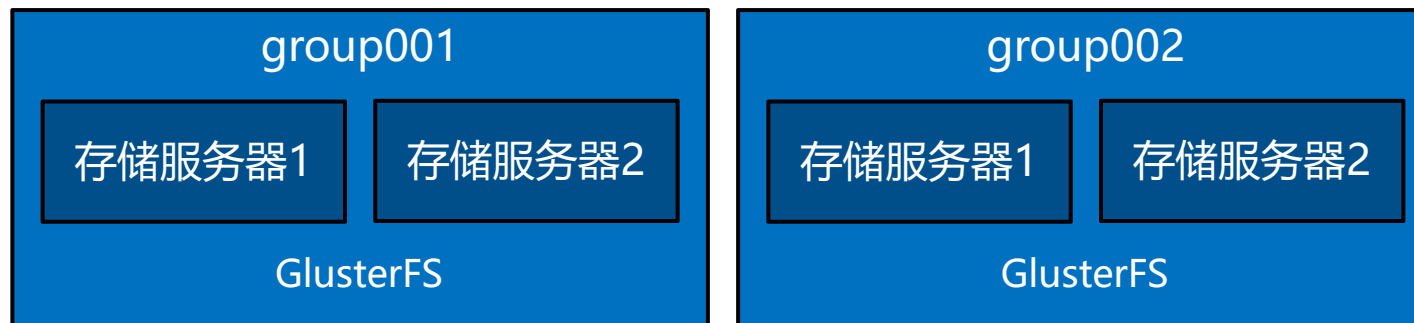
- 跟踪服务器数据库(tnv\_trackerdb)
  - 组信息表(t\_groups\_info)
  - 组路由表(t\_router)
- ID服务器数据库(tnv\_idsdb)
  - ID生成表(t\_id\_gen)
- 存储服务器数据库(tnv\_storagedb)
  - 文件信息表(t\_file\_01)
  - 文件信息表(t\_file\_02)
  - 文件信息表(t\_file\_03)



# 组信息表(t\_groups\_info)

id	group_name	create_time	update_time
1	group001	2020-11-01 12:53:56	2020-11-01 12:53:56

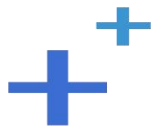
- t\_groups\_info表中记录了全部可用的组名，在存储服务器的配置文件中指明该存储服务器所隶属的组，其名称必含在此表之中



# 组路由表(t\_router)

id	appid	userid	group_name	create_time	update_time
1	tnvideo	tnv001	group001	2020-11-01 12:55:03	2020-11-01 12:55:03

- 跟踪服务器在处理来自客户机的，获取存储服务器地址列表的请求时，先查询 t\_router表，根据用户ID(userid)获取其对应的组名(group\_name)，再于该组中查找处于活动状态的存储服务器
- 若t\_router表中没有与该用户对应的组，则从t\_groups\_info表中获取全部组名，随机抽取一个，分配给该用户，并记入t\_router表



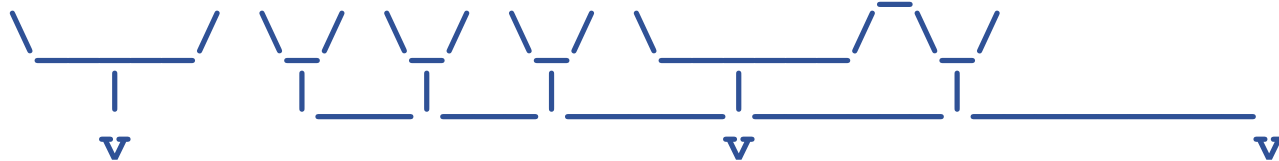


# ID生成表(t\_id\_gen)

id	id_value	create_time	update_time
../data	100	2020-11-01 12:55:03	2020-11-01 12:55:03

- ID服务器采用键(id)值(id\_value)对的形式管理保存在每个存储路径下的文件的ID号。该ID号将参与生成文件在存储路径下的各级子目录和文件名，如：

- ../data/000/000/000/5F9E3FA7\_000



存储路径

时间戳

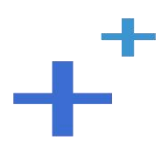
源自ID服务器生成的文件ID

- 针对与每个键相对应的ID值，ID服务器采取内存累加，数据库步进的方式，兼顾唯一性和效率

内存: 0 1 2 ... 99 100 101 102 ... 199 200 201 202 ...

库表: 100 \_\_\_\_\_ / \_\_\_\_\_ 200 \_\_\_\_\_ / \_\_\_\_\_ 300 \_\_\_\_\_ / \_\_\_\_\_

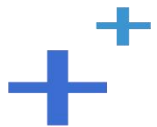
知识讲解



# 文件信息表(t\_file\_01/02/03)

id	appid	userid	status	file_path	file_size	create_time	update_time
5f9e3fa70 85e5aa253 87c8cf7b5f 29000117	tnvideo	tnv001	0	../data/000/ 000/000/5F 9E3FA7_000	734529655	2020-11-01 12:55:14	2020-11-01 12:55:14

- t\_file\_xx表记录了文件ID(id)和路径(file\_path)及大小(file\_size)的对应关系
- 此文件ID并非由ID服务器生成，而是由客户端提供的用于访问该文件的标识



# 文件信息表(t\_file\_01/02/03)

- 为了保证其唯一性，在算法中包含了用户ID、进程ID、线程ID、时间戳、计数值和随机量等信息

— 用户ID 进程ID 线程ID 随机数



| MD5



时间戳 数字摘要 计数值 随机数

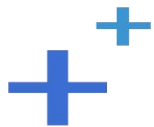


文件ID

- 不同用户上传的文件会被记录在不同的表中，表名的最后两个字符，由用户ID的哈希值计算得到：t\_file\_01、t\_file\_02和t\_file\_03

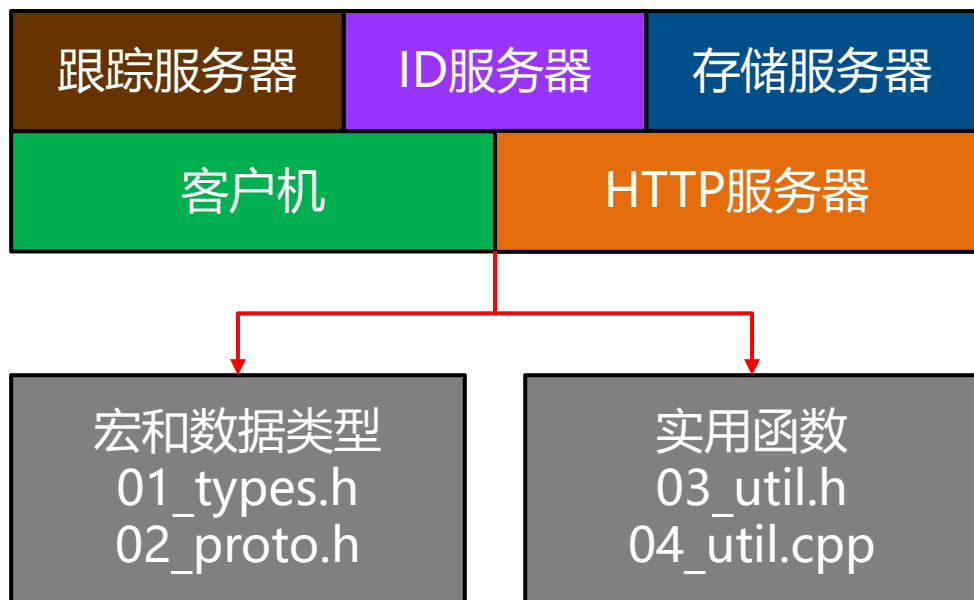


# 模块设计



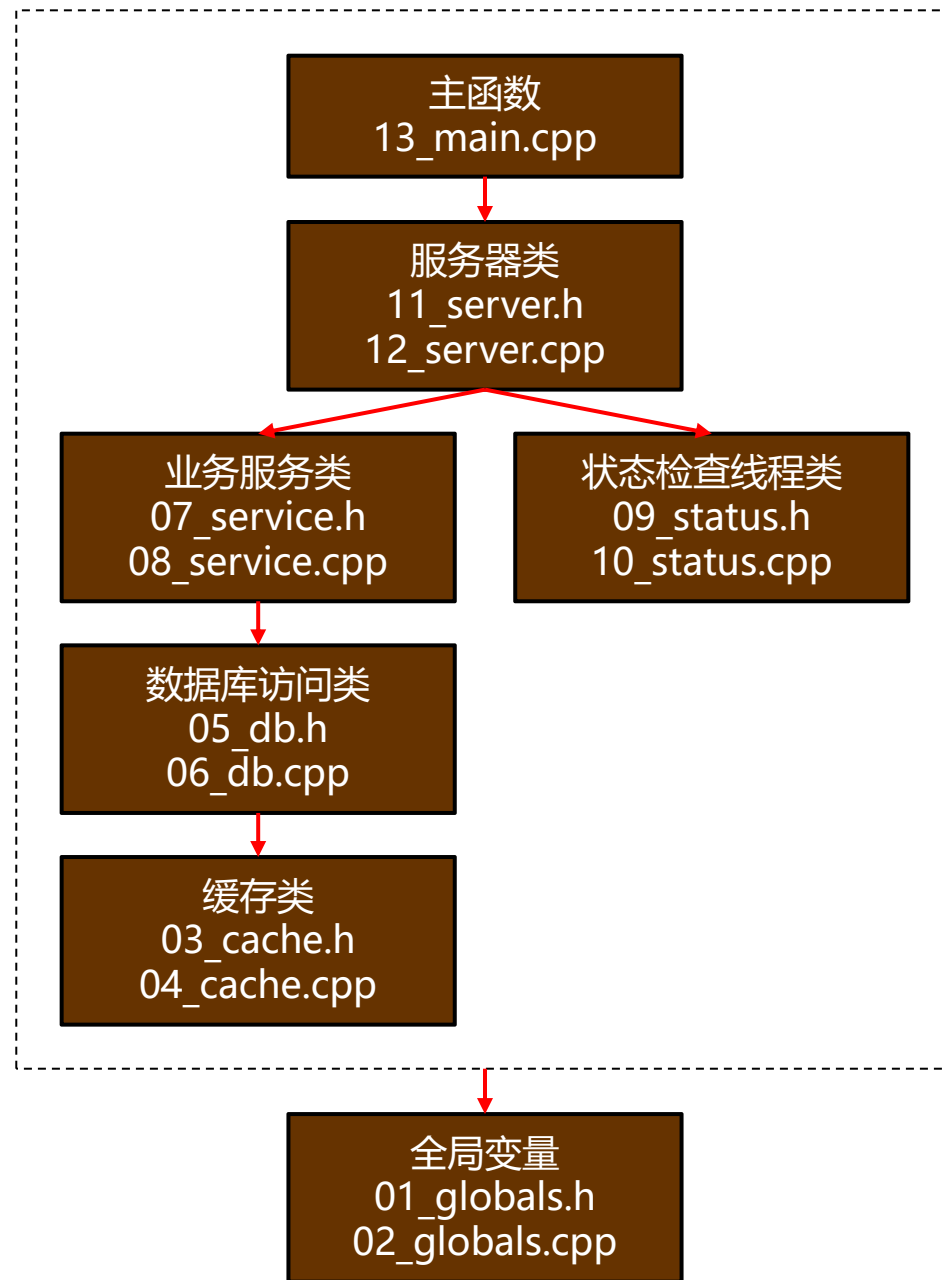
# 公共模块

- 01\_common



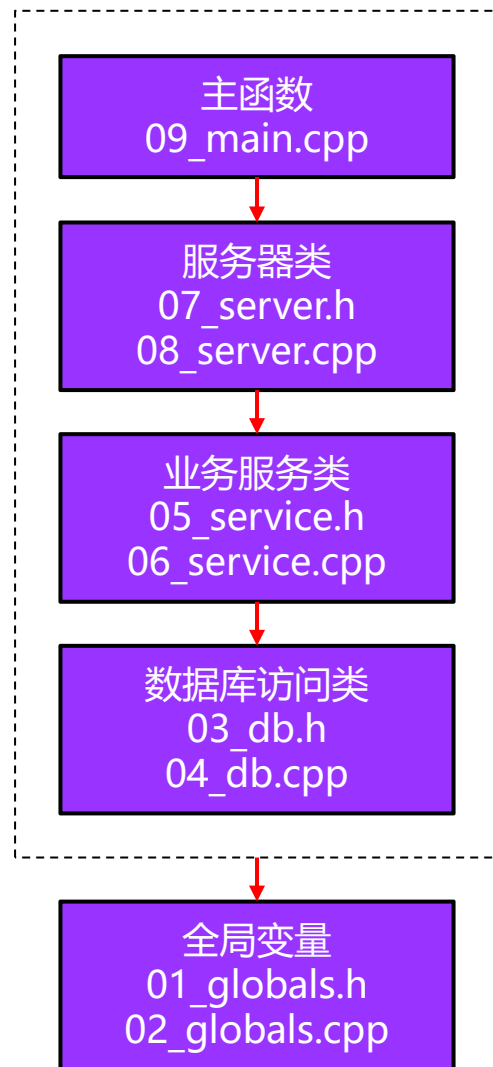
# 跟踪服务器

- 02\_tracker



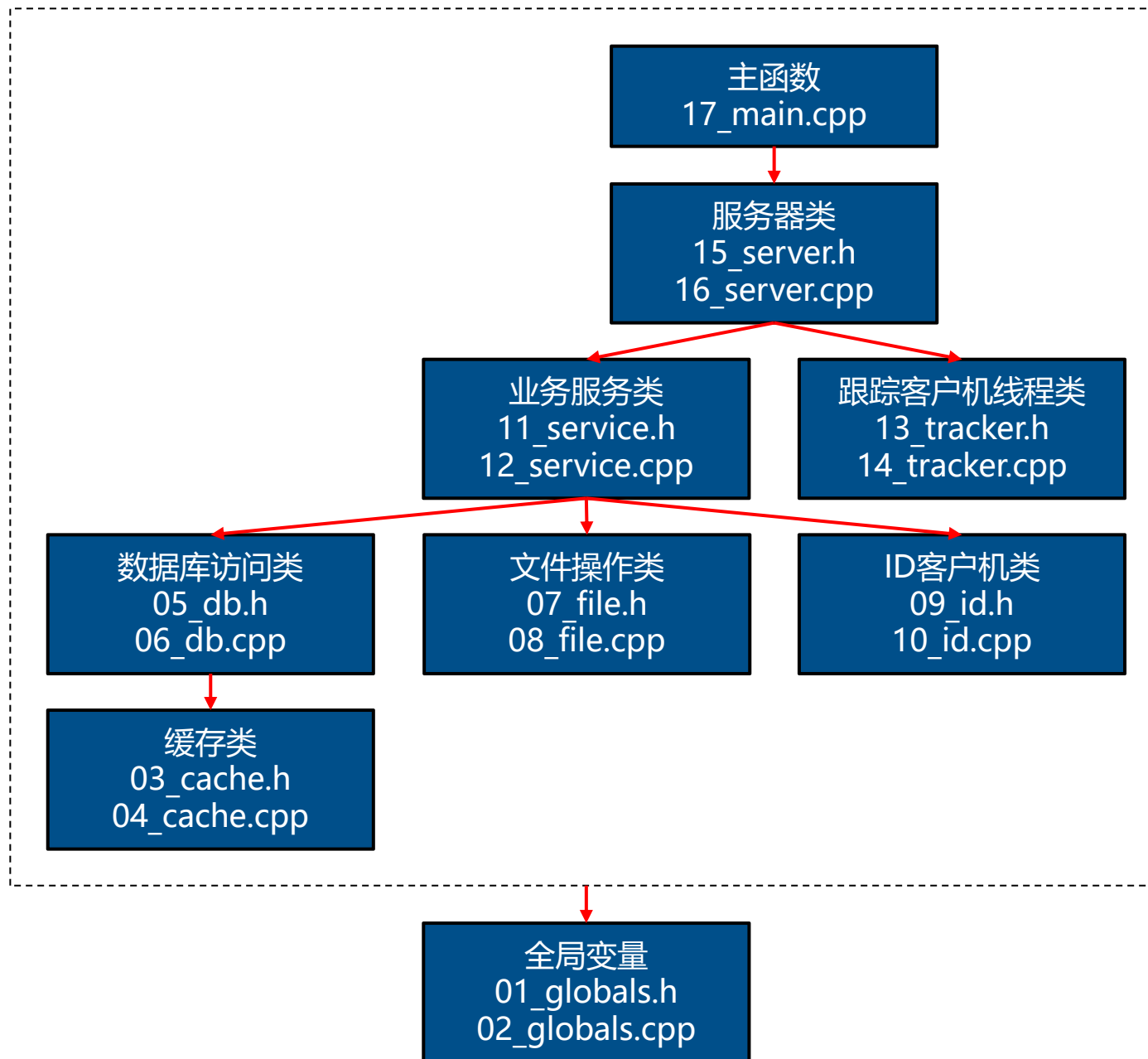
# ID服务器

- 03\_id



# 存储服务器

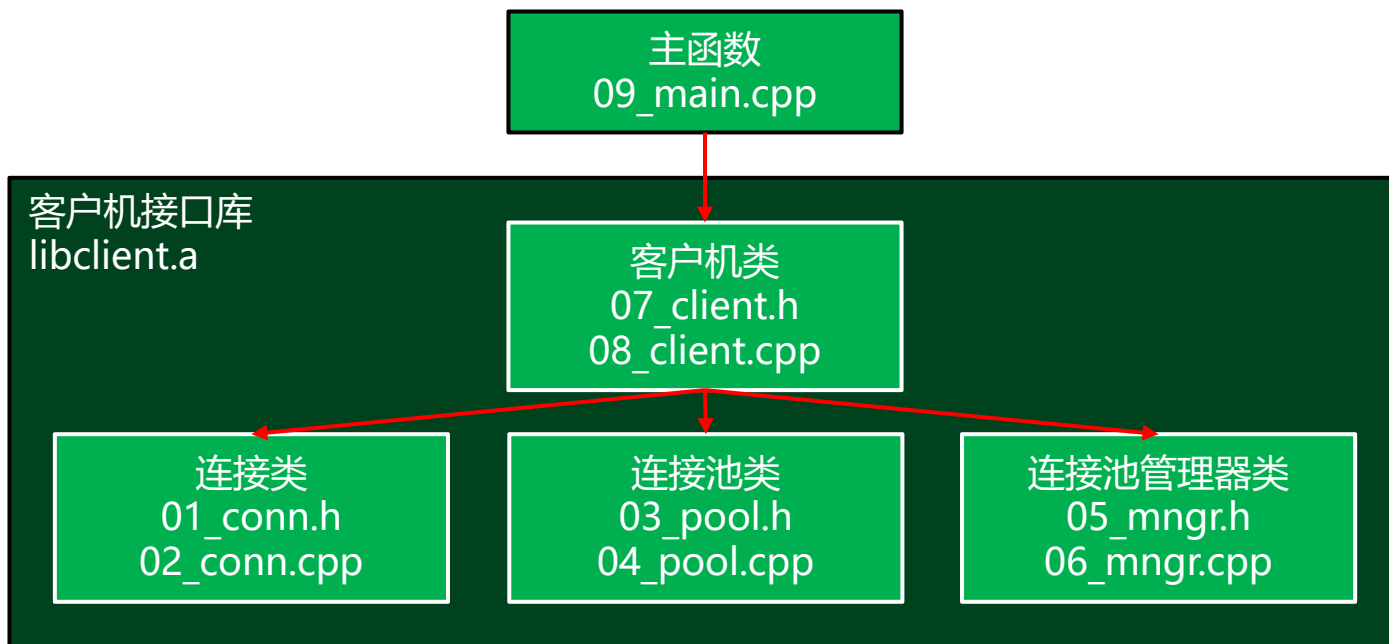
- 04\_storage





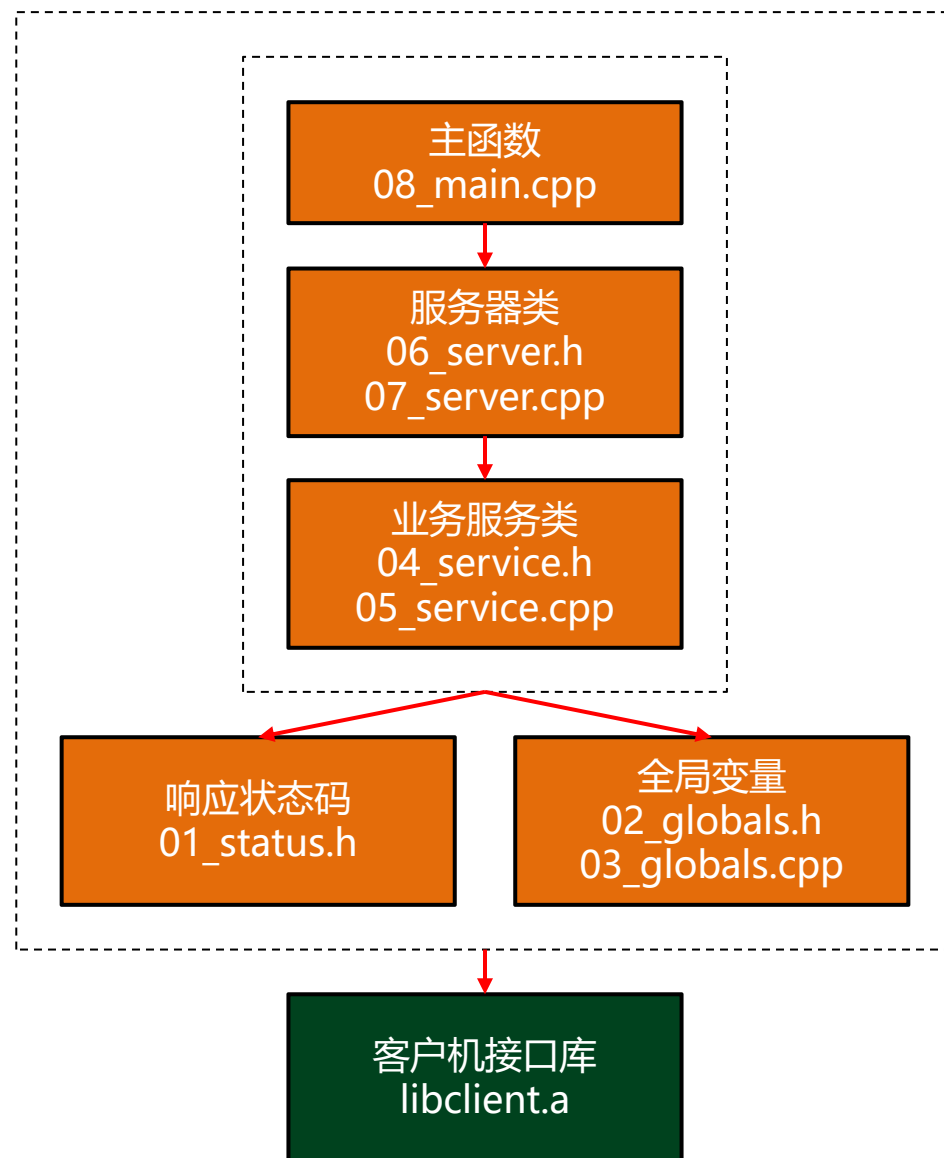
# 客户机

- 05\_client



# HTTP服务器

- 06\_http



# 复习课见