

---

# tinyFTP

---

64

commits

21

days

文件服务器和客户端

Wenchy

# 前言

- ✓ 单个文件 (**6G/2T**) 的上传下载
- ✓ 目录项分页显示
- ✓ 整个目录的上传下载 (递归, 迭代)
- ✓ 用户空间隔离 (**CWD, sqlite3**) , 多线程并发服务
- ✓ 断点续传 (**Slice + Filesize + MD5, sqlite3**)
- ✓ 极速妙传 (**Filesize + MD5 + HardLink, sqlite3**)



# Packet设计与交互模型

基本功能（文件和目录的上传下载等）

特色功能（断点续传和极速妙传）

总结与展望

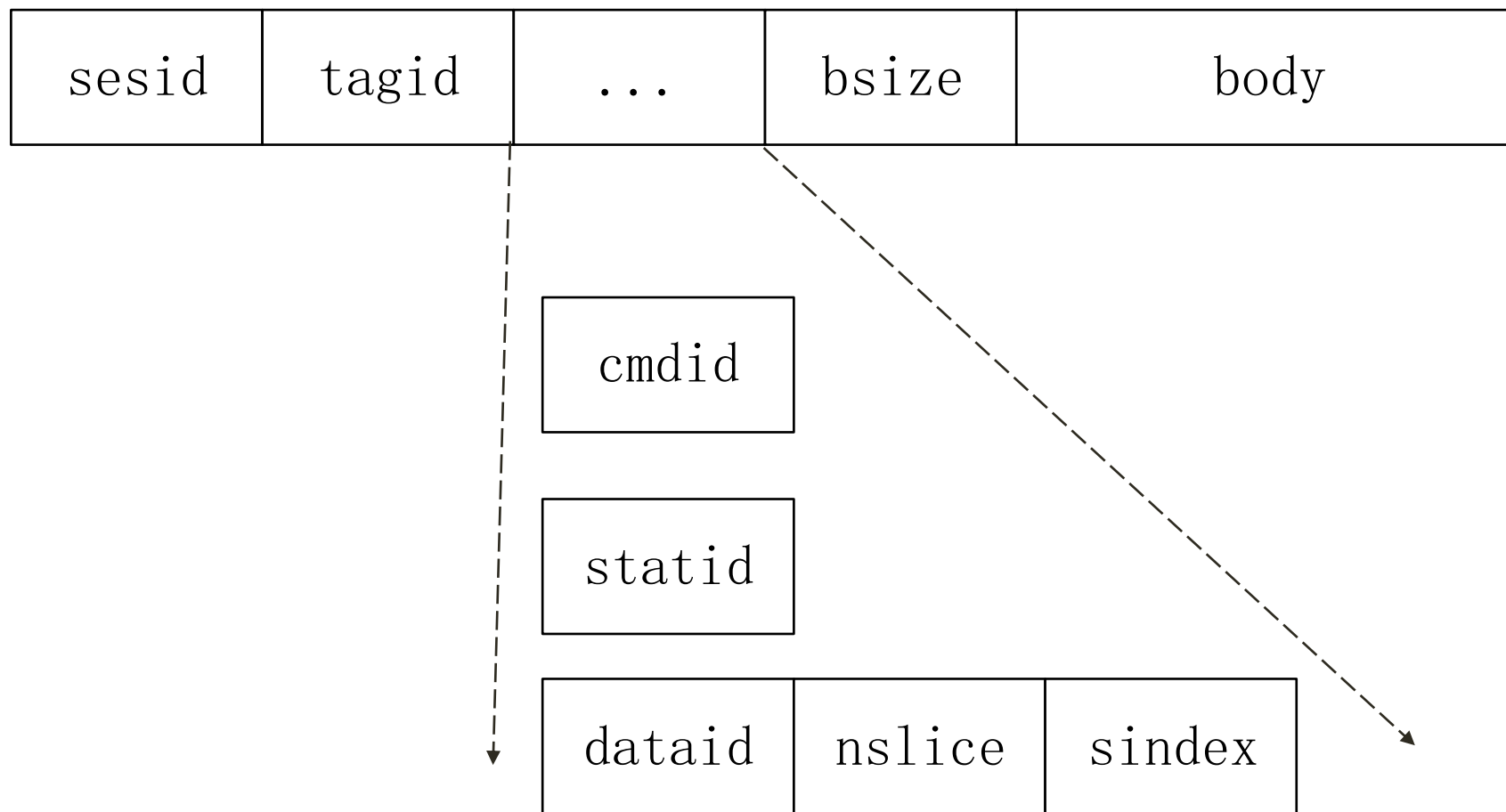
# tinyFTP

PACKET设计与交互模型

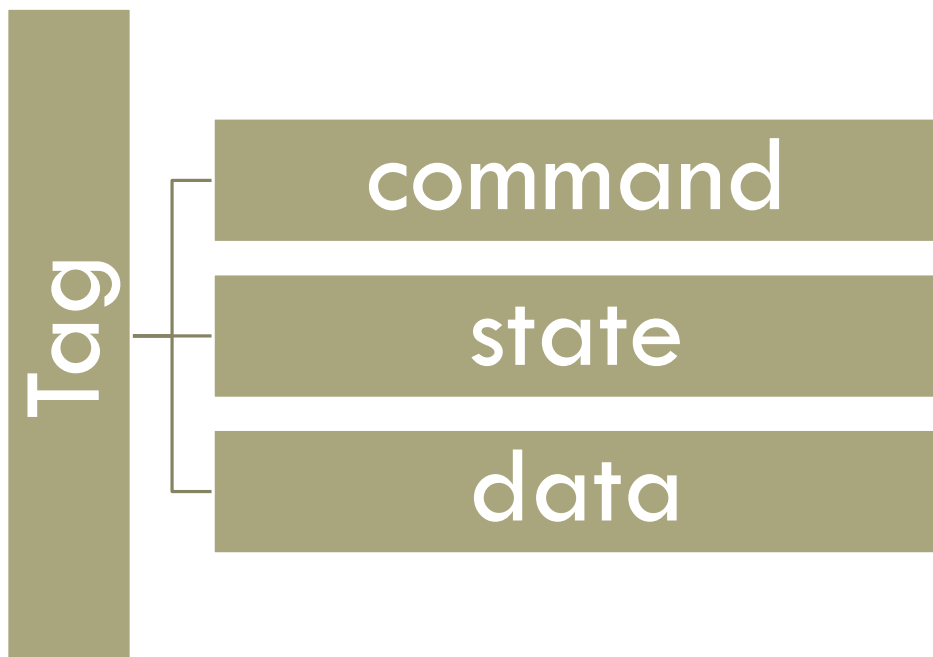
Tag

Command  
Status  
Data

# PACKET 格式



# 三种TAG类型



基本命令：PUT, GET, LS, RGET, RPUT等

事务状态：OK, ERR, TERM, EOT, EOF等

数据类型：FILE, LIST, TEXT等

## Tag

```
typedef enum tagID
{
    // command
    TAG_CMD = 1,
    // status
    TAG_STAT,
    // data
    TAG_DATA
} TagID;
```

## Command

```
typedef enum cmdID
{
    USER = 1,
    PASS,
    USERADD,
    USERDEL,
    GET,
    PUT,
    LS,
    CD,
    RM,
    PWD,
    RGET,
    RPUT,
    ...
} CmdID;
```

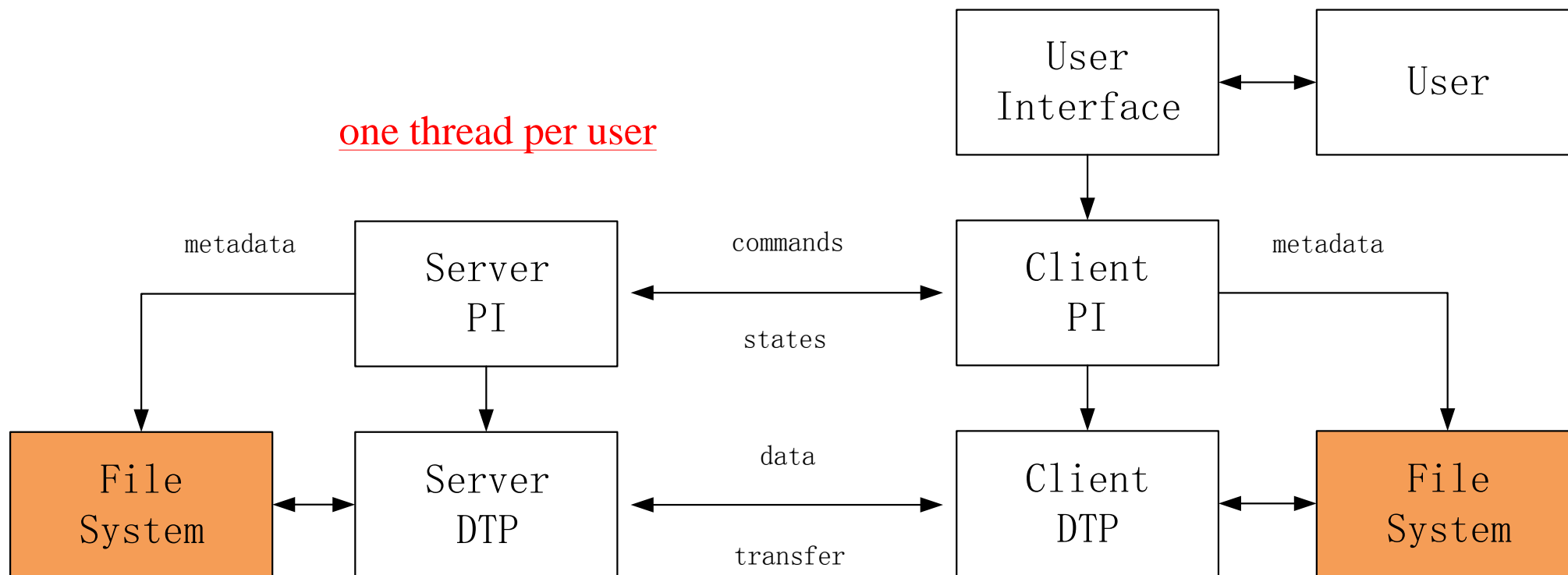
## State

```
typedef enum statID
{
    STAT_OK = 1,
    STAT_BPR, // breakpoint resume
    STAT_CFM, // confirm
    STAT_MD5, // md5sum
    STAT_PGS, // progress
    STAT_FAIL, // fail
    STAT_ERR, // error
    STAT_CTN, // continue
    STAT_TERM, // terminate
    STAT_SIZE, // size
    STAT_WAIT, // wait
    STAT_EOF, // end of file
    STAT_EOT // end of transfer
} StatID;
```

## Data

```
typedef enum dataID
{
    DATA_FILE = 1,
    DATA_TEXT,
    DATA_LIST,
    DATA_OTHER
} DataID;
```

# 交互模型



**PI:** Protocol Interpreter

**DTP:** Data Transfer Process



---

# tinyFTP

---

基本功能

File  
Directory  
Directory Entries  
CWD

# 用户空间隔离

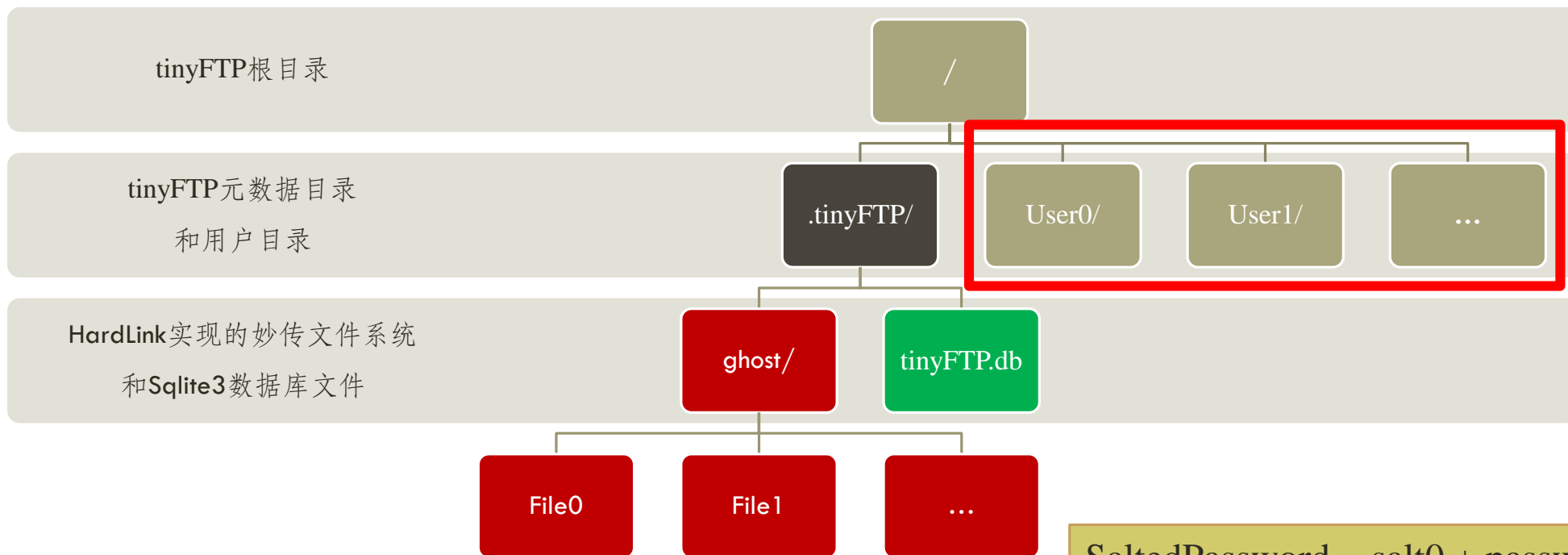
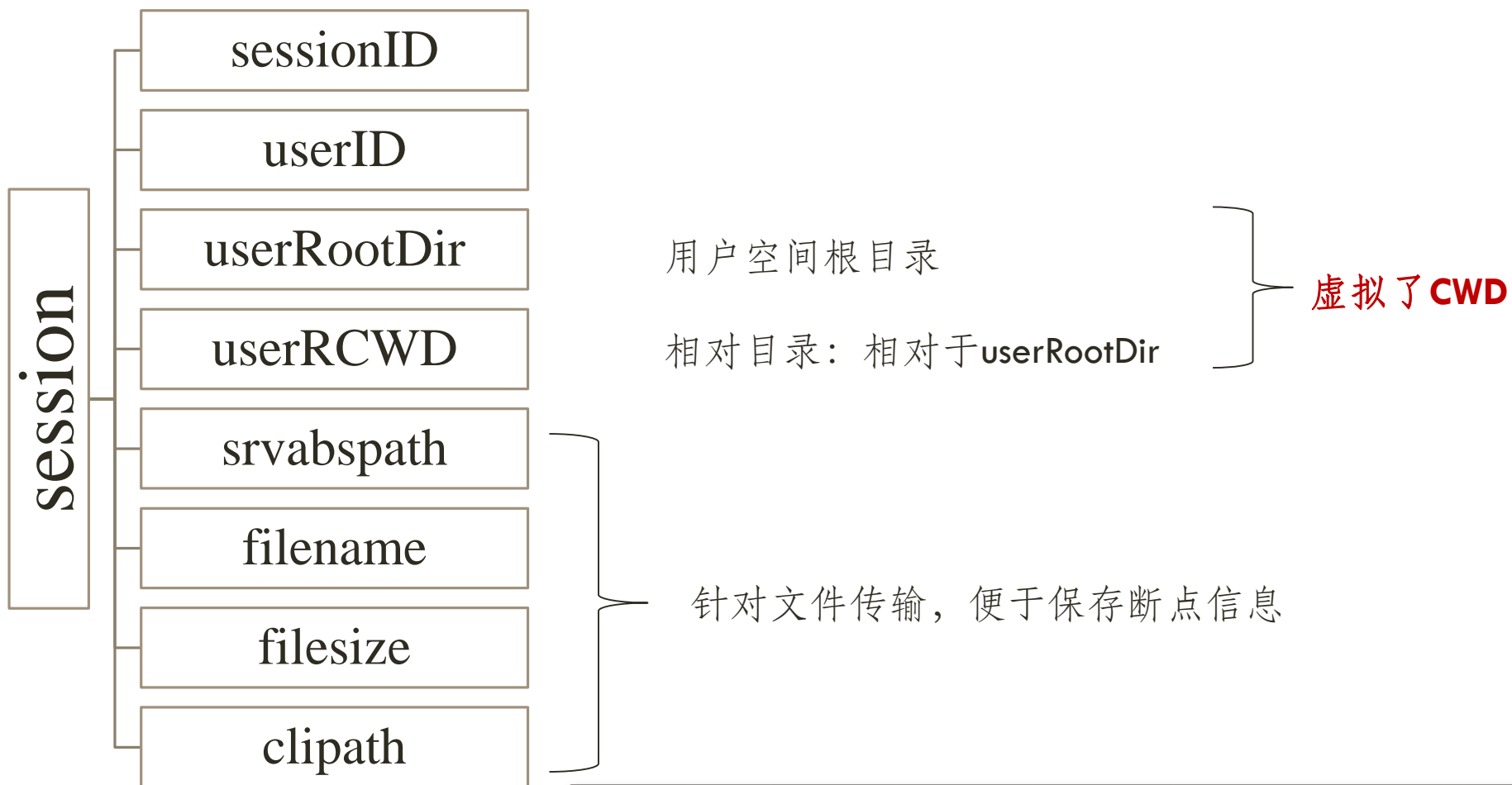


Table **USER**

ID	USERNAME	PASSWORD	RCWD	CREATE_AT	UPDATE_AT	STATE	...

$\text{SaltedPassword} = \text{salt0} + \text{password} + \text{salt1}$   
 $\text{USER.PASSWORD} = \text{MD5}(\text{SaltedPassword})$

# SESSION



一旦连接正常或意外关闭，需要持久化session以及上一个Packet

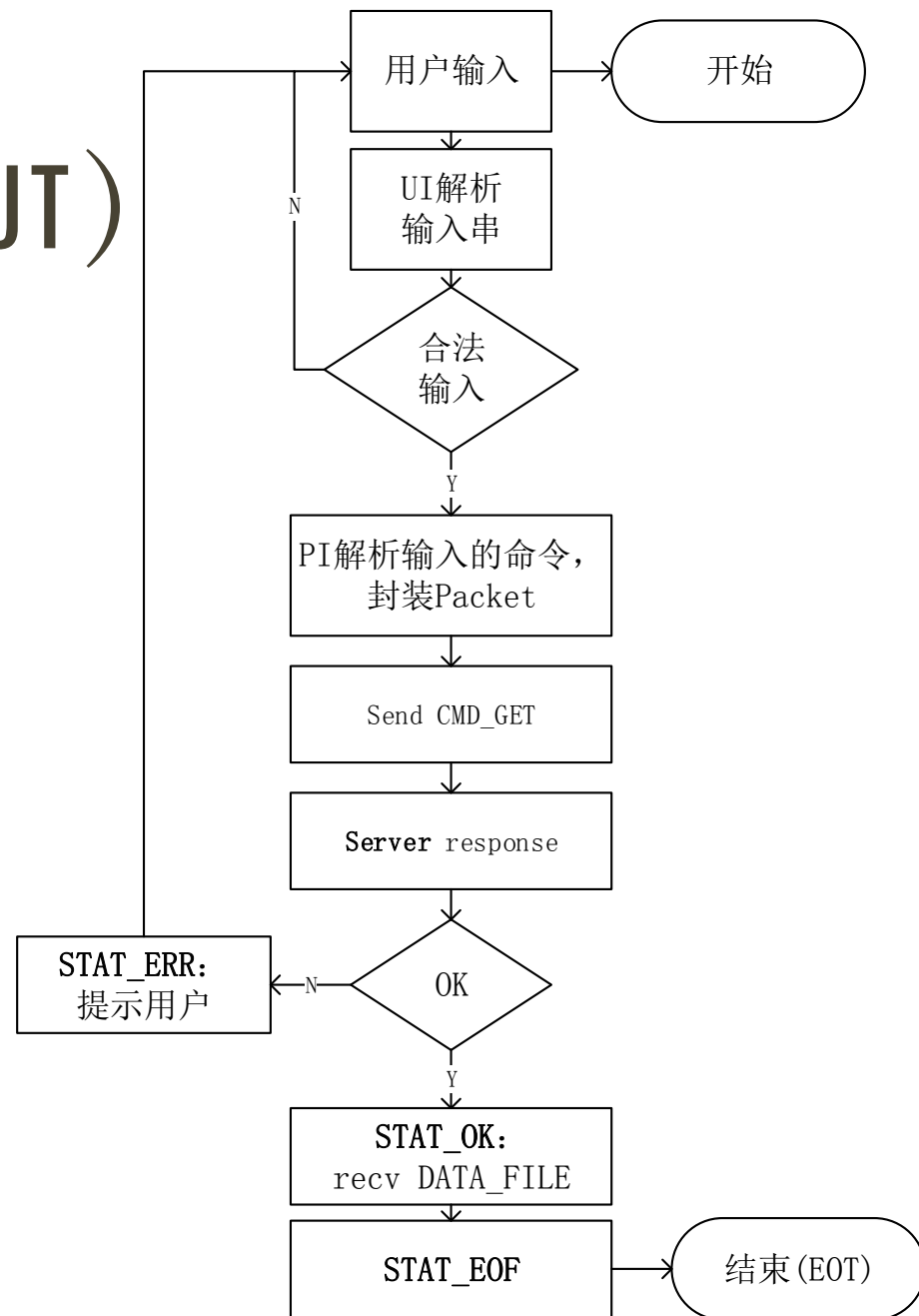
# 单个文件的传输 (GET PUT)

实测大小: 6G

理论上限: 2T

注意点:

1. 文件名已存在, 提示用户是否覆盖
2. 上传与下载进度显示的不同: **select**
3. 磁盘空间检测, 文件加锁



# 目录项列表分页 (LS)

目录项成千上万怎么办？

一页一页的传输，和用户进行交互

# 目录传输 (RGET RPUT)

递归or迭代?

迭代更友好：层次清晰有序，先建立文件夹，然后传输此文件夹内的文件

实现原理?

复用PUT, GET, MKDIR基本命令

RPUT, RGET遍历文件夹后发出上述指令，控制指令流，有序发出相应的指令

# 其它基本功能

1. MKDIR      创建目录
2. RMDIR      删除目录及文件
3. RM          删除文件
4. CD          改变当前工作目录
5. PWD        查看当前工作目录
6. QUIT       退出
7. HELP       帮助
8. 针对本地的命令：LCD，LLS等

---

# tinyFTP

---

特色功能

Breakpoint Resume  
Flash Transmission



# 断点的记录

## 统一IO:

sendOnePacket(...)      所有数据的发出  
recvOnePacket(...)      所有数据的接收

## IO复用, select:

及时发现异常, 记录断点, 保存session状态

Table **IFILE** ( interrupted file)

ID	USERID	ABSPATH	SIZE	MD5SUM	NSLICE	SINDEX	VALID	...

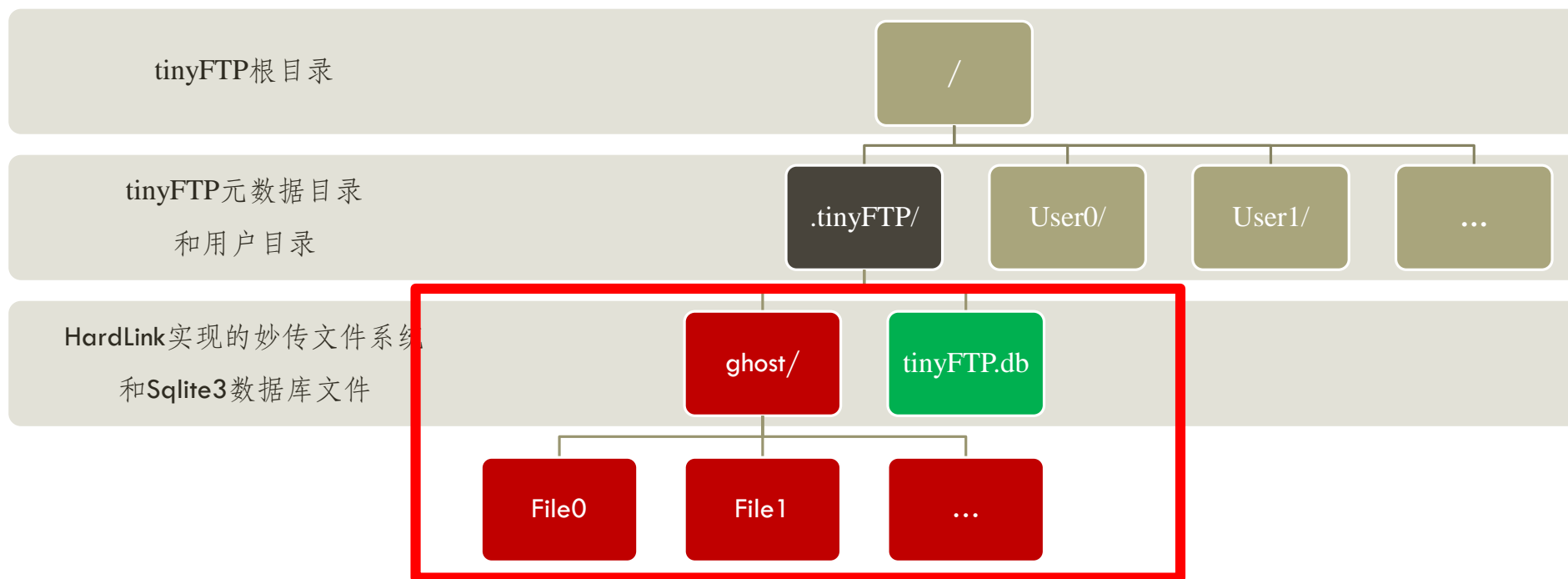
# 断点续传



Table **IFILE** ( interrupted file)

ID	USERID	ABSPATH	SIZE	MD5SUM	NSLICE	SINDEX	VALID	...

# 极速妙传 - 实现机制



Ghost文件命名规范: **timestamp\_md5sum\_filename**

示例: 20150824101401\_1dcec5a2febfaa144c882158eea77aa7\_movie.mp4

目的: 保证唯一性和方便索引

# 极速妙传 — 流程



Table **FILE**

ID	ABSPATH	FILENAME	MD5SUM	INODE	SIZE	ACCESS	...

每上传一个服务器未记录(MD5)的文件，就link此文件到ghost目录，这样inode变为2，所以即使用户立即删除自己上传的文件（inode变为2），下次传输此文件也是妙传

---

# tinyFTP

---

总结与展望

64  
commits

21  
days

# 总结

## 实现功能：

1. 基本功能：上传，下载文件和目录，**LS**分页，用户认证，用户空间隔离等
2. 特色功能：断点续传，极速妙传

## 不足之处：

1. 断点下载还未完成
2. 网络错误和异常处理不够健壮
3. 对于一个命令的**Transaction**，状态机没有规范完全
4. 服务器日志记录没有规范化（**syslog**）
5. 只是统计了热点数据，还未实现拉入内存

# 展望

- a) 后台并发模型**epoll**: 提高并发度
- b) 数据库**Mysql**: 提高并发度
- c) 热点数据优先拉入内存: 提高下载速度
- d) 多线程下载: 提高下载速度
- e) 服务器转化为**Deamon**程序

---

# tinyFTP

---

谢谢大家

Wenchy



# 程序演示

1. USER 认证，密码加盐加密
2. PUT 进度条(select)，断点续传，ghost文件建立，查看hardlinks，极速妙传
3. RM 刚上传完就删除文件，查看ghost文件hardlink，此后上传此文件是极速妙传
4. GET 热点数据统计
5. CD 用户空间隔离，虚拟CWD
6. RPUT
7. RGET