

## 1 EFH 是什么？

EFH (Extreme Feeds Handler): 多核硬件技术铸造的纳秒级行情转发服务设备, 采用了 FPGA 硬件加速技术、以及 UDP 组播模式实现的极速行情转发系统。

## 2 EFH 与 Guava 的区别？

EFH 分为期货版与证券版, 期货版 EFH 即 Guava (Guava 其实是盛立内部的项目代号, 正式的产品名称叫 EFH 期货版)。

## 3 EFH 行情为什么速度快？

第一、通过硬件进行交易所行情数据解析, 转换速度快;  
第二、多路择优的机制, 目前最新版本支持 256 路 session, 会将最快的一路组播出去;  
第三、标准的 UDP 协议, 以组播的方式转发行情数据。

## 4 极速行情协议版本

根据单个行情包长不同, 有两个不同版本的协议, 一个是 108 字节版本、一个是 80 字节。上期所目前基本都是 80 字节版本、大商所则均为 108 字节版本。两个版本除了包长不一样, 另外 108 字节同时提供高频数据、低频数据, 而 80 字节版本仅提供低频数据。

## 5 高频数据与低频数据

高频数据包含了 seq\_id、交易所 id、通道 id、quote\_flag(行情标志)、合约代码、行情时间、时间毫秒值、最新价、最新成交量(指总量)、成交金额、持仓量、买一价、买一量、卖一价、卖一量。

低频数据包含了 seq\_id、交易所 id、通道 id、合约代码、行情时间、时间毫秒值、quote\_flag(行情标志)、开盘价、最高价、最低价、收盘加、涨停价、跌停价、结算价、虚实度。

## 6 取不到涨跌停价等低频数据怎么办？

可以通过其它渠道获取, 如其供应商的组播数据源、或者从 CTP、REM 等柜台行情中去获取涨跌停价等低频数据。

## 7 关于 quote\_flag

### (1) 协议文档中描述

quote_flag	char	1	<p>0 无 time sale, 无 lev1</p> <p>1 有 time sale, 无 lev1</p> <p>2 无 time sale, 有 lev1</p> <p>3 有 time sale, 有 lev1</p> <p>说明: timesale 与 lev1 对应这字段说明</p> <p>time sale 对应字段: last_px last_share total_value total_pos</p> <p>lev1 对应字段: bid_px bid_share ask_px ask_share</p> <p>注意事项: 此字段的值是十进制方式, 而非字符方式(不能用'0', '1', '2', '3'方式来判断), 取值范围是 0 - 255</p>
------------	------	---	--

### (2) 补充说明

第一、flag 为 0 的情况, 表示某合约在某行情时间节点上, 及没有新的市场成交发生、盘口也没有变化, 交易所为了节省网络带宽会只报一个时间戳出来;

第二、最新 80 字节版本的行情, 已经将 flag=0 的数据过滤掉了, 不再组播出去;

第三、flag=1,2,3 的业务含义:

quote_flag 值	业务含义
1	某合约在该行情时间节点上总成交量增加了, 但盘口没有变化
2	某合约在该行情时间节点上总成交量不变, 但盘口有变化
3	某合约在该行情时间节点上总成交量增加了, 盘口也发生了变化

最后, 需要注意下, quote\_flag 的数据类型为 char, 程序里面如果需要打印出来, 要做一下强制类型转换, 转成 int 型打印 (才会出来 1,2,3 的值)。

## 8 大商所持仓量字段

上期所的持仓量 total\_pos 为 double 类型 (8 字节), 而大商所为 int 类型, 取低 4 字节, 在使用大商所极速行情时, 需要注意。

## 9 在 Linux 下 tcpdump 抓不到 EFH 组播出来的数据

问题排查方法:

第一、检查本机网卡配置，确定在 EFH 组播的 local 段已经分配了 IP 地址、并且接线正常;

第二、检查 tcpdump 的参数是否正确，如是否指定对了网卡等;

第三、检查/etc/sysctl.conf 配置文件

需要将 net.ipv4.conf.all.rp\_filter 和 net.ipv4.conf.default.rp\_filter 设置为 0，以解决 Linux 下多网卡收多播的问题。需要注意，修改完配置后，要重启机器。如果不重启想立即生效，可以通过执行：

sysctl -w net.ipv4.conf.all.rp\_filter=0 命令来直接修改内核中的变量值。但该命令不会修改 sysctl.conf 配置文件中的值，配置文件仍然需要修改。

## 10 在 Linux 下 tcpdump 能抓到包，但应用程序收不到数据

问题排查方法:

第一、检查防火墙是否关闭，需要关闭防火墙;

第二、检查 SELinux 状态，需要关闭

将/etc/selinux/config 配置文件中设置 SELINUX=disabled 即可，改完需要重启机器生效。如果要不重启机器立即生效，可以执行 setenforce 0 命令，临时关闭。

第三、仔细检查应用程序中各参数是否输入正确，如组播地址、端口、本地地址（必须要填可以收组播数据的 local ip）。

## 11 Windows 下收不到组播数据

虽然大部分用户，都是使用 Linux 的，也存在个别用 windows 的用户。在 windows 接收 EFH 组播数据，第一步需要配置路由（route add 命令），配完以后可用 route print 命令查看结果。

如果加完路由，应用程序仍然收不到组播数据，问题排查方法:

第一、使用 WireShark 等抓包工具，进行抓包测试;

第二、关闭防火墙;

第三、检查应用程序配置的组播参数。

## 12 关于丢包

客户端接收数据是否存在丢包现象，可以通过sequence是否连续做判断。造成丢包的情况:

第一、服务器端输出数据就缺了（这种情况是行情系统的问题，跟客户端无关);

第二、网络上面的丢包;

第三、应用程序自身丢包。这种情况出现的概率最大，原因是应用程序接收处理数据的速度不够快，如采用同步方式记录行情日志也可能造成丢包。

## 13 关于 demo 程序

盛立提供了行情 demo 程序，包括：

demo	业务含义
guava_demo	108字节行情的普通demo程序，通过recvfrom()函数基于socket实现
guava_short_demo	80 字节行情的普通 demo 程序，通过 recvfrom()函数基于 socket 实现
efh_sf_api_demo	基于 solareflare 的 ef_vi 模式接收数据的 demo 程序。

注意下基于 solareflare 实现的 Kernel bypass 方式接收数据，并非属于盛立产品所提供的功能，我们只是提供了一个样例程序，用户可以自己研究这一块的实现。

## 14 OpenOnload 与普通 socket 程序

在同一台服务器上面，不能同时使用 OpenOnload 方式与普通 socket 方式（走内核内核 TCP/IP 协议栈）同时接收行情数据。如果已经使用了 OpenOnload 方式，普通 socket 方式会收不到行情数据。

## 15 UDP 包头

cksum 值均为 0.