

调度系统中的信令分发器^①

赵红民^{1,2}, 李鸿彬², 申奇³, 李政芝⁴

¹(中国科学院大学, 北京 100049)

²(中国科学院沈阳计算技术研究所, 沈阳 110168)

³(中国电信股份有限公司北京研究院, 北京 100035)

⁴(辽沈工业集团有限公司, 沈阳 110045)

摘要: 在研究了 OpenSIPS 开源项目之后, 设计并实现一个调度系统中的信令分发器, 以解决目前调度系统中的系统规模收缩等问题. 介绍了调度系统的应用及研究现状, 分析了 OpenSIPS 的具体情况, 描述了信令分发器的功能, 阐述了信令分发器的设计与实现, 最后通过实验验证了分发器的可用性.

关键词: OpenSIPS; 信令分发器; 调度系统; SIP

Signal Dispatcher in Dispatch System

ZHAO Hong-Min^{1,2}, LI Hong-Bin², SHEN Qi³, LI Zheng-Zhi²

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology of Chinese Academy of Sciences, Shenyang 110168, China)

³(China Telecom Corporation Limited Beijing Research Institute, Beijing 1000168, China)

⁴(Liao Shen Industries Group Co., Ltd, Shenyang 110045, China)

Abstract: After conducting the research about OpenSIPS, which is an open-source project, a signal dispatcher of dispatch system is designed and implemented to solve problems in dispatch system such as shrinking problem of the system scale. This essay introduces the current application and research situation of the dispatch system and analyzes its specific circumstances, and also describes the functions of signal dispatcher, illustrates its design and implementation, and verifies its feasibility through experiments in the end.

Key words: OpenSIPS; signal dispatcher; dispatch system; SIP

调度系统是工业生产和业务活动中负责指挥调度的专用通信系统. 在工业企业、交通运输、公安和消防等部门中, 为了生产及业务活动, 需要快速的传递命令和通知. 常常采用调度系统来及时的指挥和调度人力、物资等. 由于不同部门各自的业务需要, 对调度系统的性能要求也各有不同, 但从调度系统具有指挥、调度和监督等重要作用来看, 其基本性能大致相同^[1].

随着IP通信技术的发展, 功能单一基于程控的调度指挥系统已经不能满足各单位调度指挥的要求, IP调度系统已经成为未来发展的主要趋势, IP语音调度系统开始取代原来的程控调度系统, 同时IP多媒体调度通信的需求开始日益提升, 大唐高鸿推出的集语音

调度、视频调度、视频会议、视频监控和桌面触摸屏为一体的多媒体调度系统, 为煤矿企业的安全生产提供着“四合一”的整体解决方案. 世纪网通推出的 CoreX 1000 IP调度系统, 为矿井用户提供了充足的调度手段, 全方位的掌控第一线的生产、运行态势. 对于调度系统越来越多地要求对于日常生产调度和应急指挥调度系统融合, 视频接入数字化以及提供三维地图数据, 另外对于规模的要求越来越高, 特别是一些大型煤矿、电力等企业, 要求中心能够统一指挥和调度, 分级管理, 实现各个子单位的整体调配, 满足各级部门建立有效的指挥调度关系.

根据市场反馈, 大中型企业要求调度系统的规模

① 收稿时间:2013-11-08;收到修改稿时间:2013-12-16

均在上千个用户以上,而且要求调度控制台集中控制和调度,因此在未来的调度系统发展中,需要能够提供运营商级的IP调度系统,满足市场的新需求,然而,现在现存的一些调度系统很容易受到系统规模的限制,不能更好地用于大型企业,因此设计一种信令分发器,实现调度服务器和调度控制台之间的交互,具有重要的理论研究意义和实际应用价值。

1 OpenSIPS 介绍

随着通信 IP 化的发展,IP 传输的高带宽、低成本等优势使得越来越多的企业、电信运营商加快建设基于 IP 的各种通信应用。在通信协议 IP 化发展中,SIP 协议毫无争议地成为各大电信运营商构建其未来网络的基础协议,越来越多的 SIP 软件产品也不断出现在行业应用中。SIP 协议的标准化,同时也造就了一大批优秀的开源软件产品,包括 Asterisk、SipXecs、FreeSWITCH、OpenSIPS 等 SIP 服务端软件以及 X-lite、LinPhone、eyeBeam 等 SIP 客户端软件。

SIP 服务端软件中的 OpenSIPS 是一个比较成熟的开源 SIP 服务器,它不仅提供了 SIP 代理及路由功能,而且还提供了一些应用功能。它的结构比较灵活,通过脚本来实现他的核心路由功能,还可以灵活定制各种路由策略,应用于语音、视频通信等多种应用。同时它是目前最快的 SIP 服务器之一,可以用于构建电信级产品^[2]。

OpenSIPS 的运行流程主要包括:读入配置文件,获得服务器所需要的基本信息和路由逻辑;对配置文件中指定的模块进行加载并初始化;设置新的信号处理函数,对 SIGINT、SIGPIPE、SIGUSR1、SIGCHLD、SIGTERM、SIGHUP、SIGUSR2 等信号进行处理;进程池的初始化;内存分配、定时器、FIFO 服务器等子系统的初始化;I/O 监听端口初始化。

OpenSIPS 体系结构如图 1 所示。

框架的最上层是用于实现 SIP 消息路由逻辑的 opensips.cfg 脚本配置文件,在配置文件中,可以使用 Core 提供的 Parameter 和 Function,也可以使用众多 Modules 提供的 Function。下层,提供了网络传输、SIP 消息解析等基本功能。在左侧,通过相应的数据库适配器,可使用多种数据库存取数据。在这样的体系结构下,可以方便地通过增加功能 module 来添加需要的功能,而不会对原有系统造成影响。

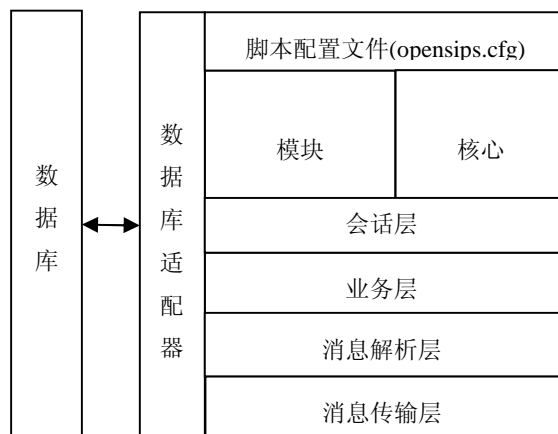


图 1 OpenSIPS 体系结构图

因此,设计一个信令分发器,作为一个模块(sigdispatch)加载到 OpenSIPS 里面,实现调度系统和调度控制台之间的交互,具有一定的实用价值。

2 信令分发器系统设计

2.1 整体架构的设计

2.1.1 功能设计

信令分发器是一个中间的实体,为控制台提供请求的转发服务。信令分发器首先提供的是路由服务,也就是说保证请求被发到更加“靠近”目标用户的地方。信令分发器在施行某些强制政策时也有用(比如,确认一个用户是否允许建立一个呼叫等)^[3]。

信令分发器的主要功能:

1. 减少控制台的连接次数,原本每个控制台都需要和调度系统有一个连接,有了信令分发器之后,控制台直接和信令分发器相连,信令分发器和调度系统相连,可显著减少控制台的连接数,降低控制台端负载。
2. 可以随时增加调度系统的个数,而无需重新启动程序。
3. 为信令定位服务器,具有一定负载均衡效果。
4. 可以使用得到信令分发器所有调度系统信息。

2.1.2 系统划分

信令分发器将控制台发来的 SIP 信令和控制信令分别发到 SIP 信令分发器和控制信令分发器,通过分发策略表为这些信令找到合适的目的地址,然后将这些信令分发到调度系统,整体结构图如图 2 所示。

信令分发器接收到消息之后,根据消息类型的不同进行不同的处理。SIP 消息处理的大致过程为:解析消息到结构体 struct sip_msg;清除所有分支,如果是请

求消息, 则执行模块注册的前脚本回调函数(在执行路由逻辑之前的函数), 执行路由引擎, 执行各模块注册的回调函数, 执行各模块注册的后脚本回调函数, 如果是响应消息, 则执行模块注册的前脚本回调函数, 执行各模块注册的回后回调函数, 转发响应消息, 执行各模块注册的后脚本回调函数, 最后释放消息结构体. 控制信令根据信令的作用的不同, 有些信令不需要经过分发表, 直接发给所有的调度系统而有些信令需要经过分发表转发到特定的调度系统. 其余信令将要被直接丢弃.

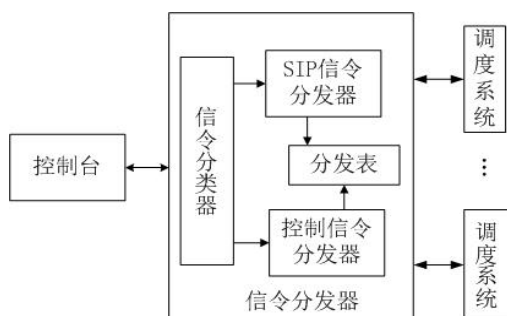


图 2 信令分发器整体结构图

信令分发器模块接口分为参数接口(Exported Parameters), 主要用来在配置文件里设置参数, 传到分发器模块. 函数接口(Exported Functions), 提供函数接口, 即在配置文件里经过配置可以直接使用的函数, 管理函数接口(Exported MI Functions)^[4].

2.2 数据结构设计

该系统采用 c 语言来完成, 主要设计的数据结构是目的地址结构体, 目的地址集合结构体及使用了大量的核心里面提供的结构体, 如消息结构体(sip_msg), 字符结构体(str)等.

主要的几个结构体:

typedef struct _ds_dest//目的地址结构体

```
{
    str uri;//目的地址
    str attrs;//uri 的属性
    int flags;
    int weight;//该目的地址的权重
    struct socket_info *sock;//套接字信息
    struct ip_addr ips[DS_MAX_IPS]; /* IP 地址 */
    unsigned short int ports[DS_MAX_IPS]; /* 请求 URI 的端口号 */
```

```
    unsigned short ips_cnt;//IP 数
    unsigned short failure_count;
    void *param;
    struct _ds_dest *next;
} ds_dest_t, *ds_dest_p;
typedef struct _ds_set//目的地址集合结构体
{
    int id; /* 该集合的 id */
    int nr; /* 该集合中地址的数目 */
    int last; /* 该集合中最后使用的地址 */
    int weight_sum; /* 该集合中目的地址的权重和 */
    ds_dest_p dlist;//目的地址列表
    struct _ds_set *next;
} ds_set_t, *ds_set_p;
```

3 信令分发器实现

信令分发器采用 c 语言来实现, 参考了 OpenSIPS 中其它模块的设计, 使用了核心模块中的大量函数和功能.

3.1 模块接口的实现

在 OpenSIPS 的模块里面需要提供一些参数, 可以在配置文件里面配置这些参数, 以使程序结构更加灵活. 模块是实现某一特定功能的属性和函数的集合^[5]. 模块的对外接口由全局变量 exports 表示, exports 的结构如下:

```
struct module_exports
{
    char* name; /* 模块名称 */
    cmd_export_t* cmds; /* 输出函数列表 */
    param_export_t* params; /* 输出参数列表 */
    init_function init_f; /* 初始化函数 */
    response_function response_f; /* 回应消息的处理函数, 本函数和以下函数可为空 */
    destroy_function destroy_f; /* 销毁函数*/
    onbreak_function onbreak_f;
    child_init_function init_child_f; /*子初始化函数*/
};
```

其中 cmd_export_t 的结构如下:

```
struct cmd_export_t {
    char* name; /* 输出函数的名称 */
```

```

cmd_function function; /* 指向输出函数的指针 */
int param_no; /* 输出函数的参数个数 */
fixup_function fixup; /* 输出函数参数的修正函数 */

int flags; /* 输出函数的标志 */
};

```

param_export_t 的结构如下:

```

struct param_export_ {
char* name; /* 输出参数的名称 */
modparam_t type; /* 输出参数的类型 */
void* param_pointer; /* 指向输出参数的指针 */
};

```

数据结构中个别函数的具体含义如下:

初始化函数: 在系统启动过程中被调用执行, 完成如注册函数、打开文件、变量赋初值等初始化操作。

子初始化函数: 在系统启动过程中且创建子进程后被调用执行, 完成如连接数据库等初始化操作。

回应消息的处理函数: 当模块需要处理回应消息时, 设置此函数, 并在模块的初始化函数中进行注册, 当接收到回应消息时, 调用此函数。比如, 在事务处理模块中, 若接收到其回应, 需重新设置计时器资源; 如果此事务设置了 on_reply 标志, 还要执行配置文件中的 on_reply 路由逻辑。

销毁函数: 在系统退出过程中被调用执行, 用来释放初始化函数占用的资源。

3.2 SIP 信令分发器的设计实现

SIP 消息有从客户机到服务器的请求消息(Request)和从服务器到客户机的响应消息(Response)两种类型。请求消息主要有发起呼叫(INVITE), 响应确认(ACK), 释放连接(BYE), 在连接建立起来之前发送, 主叫取消呼叫(CANCEL), 在注册服务器上注册用户(REGISTER), 查询服务器的能力(OPTIONS)。响应消息是服务器向客户端反馈对应请求的处理结果的 SIP 消息, 包括 1XX, 2XX, 3XX, 4XX, 5XX, 6XX。

选取目的地址的算法的流程为: 解析消息, 封装成 sip_msg 格式, 根据给出的参数先选择出目的地址集合, 然后按照要求选择合适的分发算法。根据分发算法得到的哈希号, 再根据数据库里目的地址的权重等特征来选取合适的目的地址。选好地址后, 更新消息结构体信息, 如果采用轮询算法, 则需要更新目的地址集合中最后使用的地址。最后更新全局变量的值。

3.3 控制信令分发器的设计实现

控制信令用来获取调度系统初始化过程中的一些信息, 如获取群组信息, 分机列表, 通话列表等信息。获取所有调度系统的目的地址的流程图如图 3 所示, 其中 _ds_list 是目的地址的集合, rpl 是指向 struct mi_node 型的指针变量, 用来表示节点。

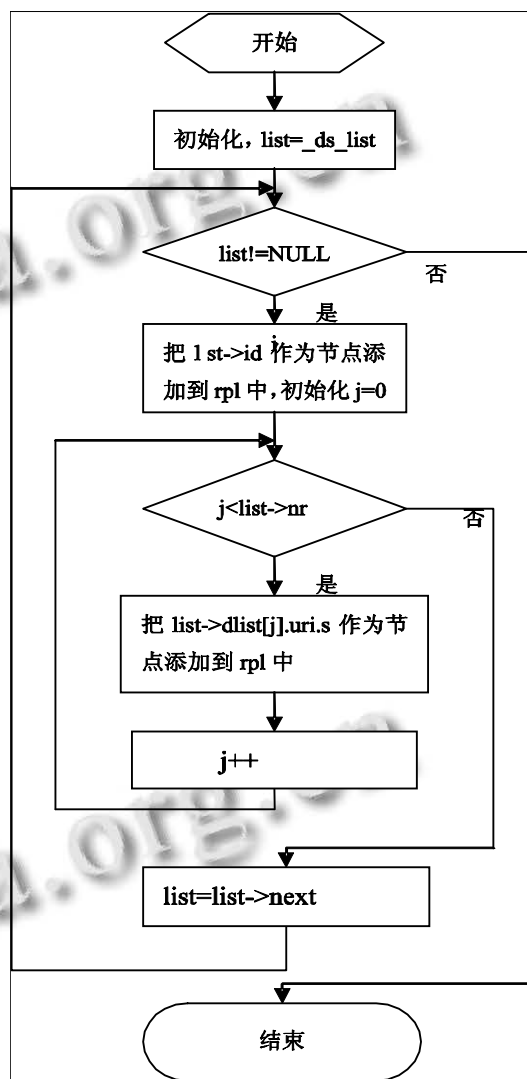


图 3 获取调度系统目的地址流程图

3.4 分发表的设计实现

根据选择的地址的方法不同, 可以选择不同的类型作为分发算法的参数, 主要有根据呼叫者的 Call-ID, 呼叫者的 uri 等哈希得到目的地址或者按照轮询调度的方法选择下一个目的地址或者采用随机算法随机选取一个目的地址。其中的一种哈希算法如下, str 结构体类型如下:

```

struct str{
Char *s;//指向字符串的指针
Int len;//字符串的长度
};
int hash(str *x)//以字符串作为哈希参数
{
    h=0;
if(x)
{
    p=x->s;
    if (x->len>=4)//如果 x 的长度大于 4
    {
        for (; p<=(x->s+x->len-4); p+=4)
        {
            v=(*p<<24)+(p[1]<<16)+(p[2]<<8)+p[3];
            h+=v^(v>>3);
        }
    }
    v=0;
    for (;p<(x->s+x->len); p++)
    {
        v<<=8;
        v+=*p;
    }
    h+=v^(v>>3);
}
return h;
}

```

该哈希算法的最大好处是计算哈希值的时候利用到了字符串中的每一个字符,且冲突较少.分发器算出信令的哈希值后,SIP信令分发器和控制信令分发器根据该哈希值选择目的地址,进行下一步的操作.

4 实验

需要在数据库中添加表 sigdispatcher,这个表用来存放目的地址,信令经过信令分发器之后按照一定的规则从数据库中选择目的地址.该实验通过信令的 Call-ID 字段选择目的地址.Call-ID 头域扮演了一个唯一的标识,来对一系列信息分组^[6].在 sigdispatcher 表中添加数据如表 1 所示:

表 1 实验数据表

id	setid	destination
1	1	j.doe@192.168.139.30
2	1	k.doe@192.168.139.40
3	1	l.doe@192.168.139.50
4	1	m.doe@192.168.139.60

实验过程中,把控制台连到 OpenSIPS 上,然后 OpenSIPS 连到 4 个调度系统,为调度系统设定地址信息如表 1 所示.实验结果如表 2 所示:

表 2 实验结果图

Call-ID	哈希号	目的地址 id
f8ea48de8@localhost.locldomain	3	4
afgtu9eoq@localhost.locldomain	1	2
zxcv1ert5@localhost.locldomain	0	1

通过实验结果可以看出,根据 Call-ID 字段得到哈希号,根据哈希号从数据库中选择目的地址,表中几个不同的 Call-ID 字段得到了不同的哈希号,说明了该哈希算法的冲突是比较少的,并且信令可以分发到选择好的目的地址,从而证明了信令分发器的可用性.

5 结语

本文通过对开源代码的分析研究,设计并实现了一个信令分发器.首先分析了信令分发器的作用,接着详细描述了信令分发器的设计和实现过程,对主要函数做了详尽分析,可以完成信令分发器的基本功能,后续仍需要进一步解决控制信令和 SIP 信令交互过程,从而使调度系统有进一步的应用.

参考文献

- 1 刘沙,周长春.IP-PBX 调度系统在煤矿井下的应用.计算机系统应用,2010,19(12):147-150.
- 2 使用 openser 构建电话通信系统. <http://www.doc88.com/p-53176114761.html>.
- 3 <http://www.asterisk.org/>.
- 4 <http://www.opensips.org/Vilas>.
- 5 张伟.SIP 预付费代理服务器的设计和实现[硕士学位论文].沈阳:中国科学院沈阳计算所,2005.
- 6 Rosenberg J, Schulzrinne H, Camarillo G, et al. SIP: Session Initiation Protocol. <http://www.ietf.org/rfc/rfc3261.txt?number=3261>.
- 7 SIP:Session Initiation Protocol. IETF RFC 3261,2002.